Multichannel Cross-Layer Routing for Sensor Networks

Noradila Nordin

A dissertation submitted in partial fulfillment

of the requirements for the degree of

Doctor of Philosophy

of

University College London

Department of Electronic & Electrical Engineering

University College London

2017

I, Noradila Nordin, confirm that the work presented in this thesis is my own. Where information has been derived from other sources, I confirm that this has been indicated in the thesis.

© 2013-2017, Noradila Nordin

Department of Electronic & Electrical Engineering
University College London

Abstract

Wireless Sensor Networks are ad-hoc networks that consist of sensor nodes that typically use low-power radios to connect to the Internet. The channels used by the low-power radio often suffer from interference from the other devices sharing the same frequency. By using multichannel communication in wireless networks, the effects of interference can be mitigated to enable the network to operate reliably.

This thesis investigates an energy efficient multichannel protocol in Wireless Sensor Networks. It presents a new decentralised multichannel tree-building protocol with a centralised controller for ad-hoc sensor networks. The proposed protocol alleviates the effect of interference, which results in improved network efficiency, stability, and link reliability. The protocol detects the channels that suffer interference in real-time and switches the sensor nodes from those channels. It takes into account all available channels and aims to use the spectrum efficiently by transmitting on several channels.

In addition to the use of multiple channels, the protocol reconstructs the topology based on the sensor nodes' residual energy, which can prolong the network lifetime. The sensor nodes' energy consumption is reduced because of the multichannel protocol. By using the lifetime energy spanning tree algorithm proposed in this thesis, energy consumption can be further improved by balancing the energy load in the network. This solution enables sensor nodes with less residual energy to remain functional in the network. The benefits of the proposed protocol are described in an extensive performance evaluation of different scenarios in this thesis.

Acknowledgements

I would like to thank my supervisor, Dr. Miguel Rio, for the opportunity to pursue this research and his constant guidance, advice and encouragement that made this work possible.

I would also like to thank Dr. Richard G. Clegg for his help from the beginning in reshaping my research, providing advice, guidance, and countless discussions. Many thanks as well to all my colleagues in the Communications and Information Systems group.

Also, I would like to thank the external and internal examiners, Dr. Hamed Haddadi and Dr. David Griffin for their valuable comments and feedback.

Lastly, and most importantly, I would like to thank my family, in particular my dad, Dr. Nordin Yunus, for the constant reminder, advice and for inspiring me to pursue my PhD. Also, my sister, Dr. Noradina Nordin and my brother, Mohamad Adri Nordin, for the encouragement, support and for keeping me grounded throughout the PhD life.

I would also like to thank the Malaysia government for the King's scholarship that has given me the opportunity to pursue this PhD.

Dedicated to the loving memory of my mother, Fauziah Hanim Ghazali,

a smart and strong woman whom I still miss every day.

Forever in my heart.

Contents

List of Figures

List of Tables

Abbreviations

6LoWPAN IPv6 over Low Power Wireless Personal Area Network

ACK Acknowledgement packet

ACQUIRE Active Query Forwarding In Sensor Networks

APTEEN Adaptive Periodic Threshold Sensitive Energy Efficient Sensor Network

ASN Absolute Slot Number

B-MAC Berkeley Media Access Control

BoX-MAC Physical and Link Layer Boundaries MAC protocol

CADR Constrained Anisotropic Diffusion Routing

CAOF Context-Aware Objective Function

CCA Clear Channel Assessment

CF Common frequency phase

ContikiMAC Contiki MAC protocol

Cooja Contiki network simulator

CTP Contiki Collect Protocol

DAO Destination Advertisement Object

DIO DODAG Information Object

DIS DODAG Information Solicitation

DODAG Destination Oriented Directed Acyclic Graph

ELC Exponential Lifetime Cost

ELT Expected Lifetime

Energest Contiki energy estimation module

ETX Expected number of transmissions

FSELC Fully Simplified Exponential Lifetime Cost

GAF Geographic Adaptive Fidelity

GBR Gradient-Based Routing

GEAR Geographic and Energy-Aware Routing

GPS Global Positioning System

HEED Hybrid, Energy-Efficient, Distributed Protocol

IoT Internet of Things

IPv4 Internet Protocol version 4

IPv6 Internet Protocol version 6

L^{2}AM Lifetime and Latency Aggregateable Metric

LB-RPL Load Balanced Routing Protocol

LCG Linear Congruential Generator

LEACH Low-Energy Adaptive Clustering Hierarchy

LLN Low Power and Lossy Network

LMAC Lightweight Medium Access Control

LPBR Low Power Border Router

MAC Medium Access Control

MC-LMAC Multi-Channel Lightweight Medium Access Control

MCRP Multichannel Cross-Layer Routing Protocol

MiCMAC Multichannel ContikiMAC protocol

MiCMAC-BC MiCMAC with broadcast support

OF Objective Function

OS Operating System

PEGASIS Power-Efficient Gathering in Sensor Information System

Powertrace Energest power state tracking profile

QoS Quality of Service

RDC Radio Duty Cycle

Rime Contiki lightweight communication stack

ROEE Resource Oriented and Energy Efficient

RPL Routing Protocol for Low Power and Lossy Network

RSSI Received Signal Strength Indicator

SAR Sequential Assignment Routing

Sink It is a base station and also referred to as LPBR

SLIP Serial Line Internet Protocol

SPIN Sensor Protocols for Information via Negotiation

TCP Transmission Control Protocol

TDMA Time Division Multiple Access

TEEN Threshold Sensitive Energy Efficient Sensor Network

TS Time slot

TSCH Timeslotted Channel Hopping

UDP User Datagram Protocol

uIP Micro Internet Protocol

WiseMAC Wireless Sensor MAC

WSN Wireless Sensor Network

X-MAC Low Power MAC protocol

Y-MAC Energy-Efficient Multichannel MAC protocol

Publication

- N. Nordin, R. G. Clegg and M. Rio, **Multichannel cross-layer routing for sensor networks**, 2016 23rd International Conference on Telecommunications (ICT), Thessaloniki, 2016, pp. 1-6.

Chapter 1

Introduction

1.1 Context and Motivation

A Wireless Sensor Network (WSN) is an ad-hoc network that consists of sensor nodes networked in an environment without human intervention for a duration that depends on the sensors' application. WSNs support a wide range of applications such as tracking and monitoring of the environment, home, military and commercial applications.

The sensor nodes in WSNs typically use low-power radios such as IEEE 802.15.4, a relatively short-range transmission standard radio technology in the 2.4 GHz band. The standard allows transmission to occur on several different channels within this band [48]. Unfortunately, the channels used by this technology often suffer from interference [14, 88], such as Wi-Fi (IEEE 802.11)[49, 123] and Bluetooth (IEEE 802.15.1). Sensor networks have to contend with an increasing number of devices that cause this wireless interference. Organising the network topology around this interference becomes an enabler for increasing transmission efficiency. WSNs need to be able to operate reliably in the presence of such interference in order to maximise the throughput with minimal energy costs, since deployments can be for weeks, months or longer.

Multichannel communication in wireless networks can alleviate the effects of interference, which, as a result, can improve the network efficiency, stability and link reliability, minimise latency [119] and overall energy consumption. It also enables communication between physically proximate nodes to occur simultaneously, without the risk of collision, when the communicating nodes use different channels. However, not all channels are free from interference. Some channels perform better than other channels depending on the current location and network environment. Therefore, nodes should consider hopping to another channel when the channel performance starts to decline. This also affects the sensors, and therefore network lifetime, as energy is wasted in retransmissions and packets lost in the interference environment.

1.2 Problem Statement

The vision of the Internet of Things has to accommodate wireless devices that cannot be directly connected to the Internet, which have to use ad-hoc sensor networks to be able to reach a connected node. This can happen in two main scenarios: when the connected node is out of reach of the transmission range of the sender (e.g. in wide area environmental monitoring applications) or when the primary communication channel goes down and the sensor nodes themselves need to provide the backup route.

Using a multichannel protocol has an advantage in these scenarios, as it allows communication to take place on different channels of the frequency spectrum and allows several communications to run simultaneously without the risk of collisions. Several previous studies have developed multichannel Medium Access Control (MAC) layer protocols, but, despite the potential benefits, none are widely implemented yet in real-world deployments. The energy consumption of the sensor nodes during communications is also studied as multichannel protocol reduces the energy usage in transmissions and receptions as an effect of reliable communications. However, as sensor nodes have limited computational power, prior studies opted for standard static channel allocation. The work in this thesis uses dynamic channel hopping, which transfers the channel allocation processes to a centralised controller that has the computational power and ability to run the algorithm.

In addition to MAC protocols, the routing protocols are also extensively studied to ensure efficient transmissions in WSNs. While existing routing protocols reduce the energy consumption owing to intelligent route selection, the routing protocols should also consider the benefits of routing over multiple channels, which can further improve the network's energy usage while maintaining a high rate of successful communications.

The objective of the work in this thesis is to overcome the interference problem while maintaining a maximum throughput and increasing the sensor nodes' lifetime in WSNs by introducing an energy-efficient multichannel protocol. This work takes into account all available channels to utilise the spectrum and minimise the energy consumption in transmissions by avoiding channels with interference. This thesis studies the cross-layer dimension of the protocol to take advantages of the properties in the different layers, whereas previous protocols would only concentrate on a specific individual layer, mainly the MAC layer or the network layer. This work studies the interaction between the different layers to allow improvement of the network's communication on the stack.

This approach could be implemented in the real world for environmental applications such as the monitoring of air pollution in big cities, to analyse the concentration of harmful gasses due to pollution. An energy efficient multichannel cross-layer routing protocol (MCRP) can ensure that urgent alerts are successfully transmitted as soon as possible and that monitoring data is sent periodically.

1.3 Contributions

An important aspect of this work is improving the WSN's performance by implementing a multichannel routing protocol to investigate the effect of multichannel in comparison to a single channel protocol in a noisy environment. There are several research challenges in a multichannel protocol design in terms of the synchronisation, topology formation and maintenance, as well as the initiation and selection of channels and the energy consumption of the multichannel protocol.

The multichannel protocol in this work consists of centralised and decentralised parts, where the centralised node controls the network and the decentralised node carries out the multichannel processes. The main benefit of this protocol is that it enables real-time interference detection in order to select a better channel for transmissions, which adapts to any location without the need to know the information beforehand. A multichannel protocol helps to reduce the number of packet retransmissions and losses, thus giving more efficient energy usage during communications (less energy waste). To further improve the network in terms of the lifetime, the topology is reconstructed by maximising the energy of the minimum energy nodes.

The work in this thesis investigates the channel selection processes based on the cross-layer dimension of the protocol during decision-making and the lifetime energy spanning tree, to further improve the multichannel protocol. The contributions of this work are organised into seven main chapters, as follows.

WSN Applications, MAC and Routing Protocols: WSN applications are widely used and deployed in various locations and environments for different purposes, from attachable smart devices and home automation to detecting natural disasters. There have been many proposals in multichannel protocols at the MAC and routing layers. However, improving only one of the layers is insufficient to enable the network to have reliable and fully functional communications in a longer time period. By using a cross-layer protocol, better network efficiency can be achieved. This allows a larger range of WSN applications to be deployed with a predictable outcome owing to efficient use of both energy and transmission quality, such as in disaster detection. WSNs have to be able to relay the information to the base station reliably at the crucial time.

Multichannel Cross-Layer Routing Protocol: A new multichannel protocol, Multichannel Cross-Layer Routing Protocol (MCRP), has been developed in this thesis, which interacts between the layers based on the research done in the previous chapter on the existing proposed protocols. MCRP is proposed to enable communications on all available channels in the spectrum to avoid interference, congestion and conflict in the network. In this chapter, the strategies in channel selection, channel switching, and quality checking of the external and internal interference are discussed. A reconnection strategy is also proposed to ensure that the network remains functional if a node joins or leaves the network. The information on channel interference and network topology from the lower layer is made available to the application layer.

MCRP Implementation: The design of MCRP that was proposed in the previous chapter is implemented in the Contiki operating system for Internet of Things (IoT). The existing components in Contiki that are essential in MCRP are explained mainly on the application, network and MAC layers. The steps and MCRP processes are explained in detail for the Low Power Border Router (LPBR) and the descendant nodes in the network. The implementation of MCRP at the Sink, which is the LPBR, is slightly different than on the descendant nodes. However, both implementations use the same MCRP strategy. MCRP consists of two main parts, centralised intelligence in the LPBR and the decentralised nodes. The LPBR implements a two-hops colouring algorithm to avoid interference between physically proximate nodes trying to communicate on the same channel. The system is fail-safe in the sense that the WSN functions if the central system that assigns channels fails.

Simulation Performance Evaluation: MCRP is evaluated in the Contiki network simulator (Cooja) [85]. An existing interference model is used to evaluate MCRP performance. The performance of MCRP is compared against a single channel protocol and an existing multichannel protocol, Orchestra, in terms of the end-to-end packet delivery, the setup overhead, the channel switching and the reconnection delay. The results showed that MCRP performed better than the other protocols in the simulated interference environment.

Testbed Performance Evaluation: In addition to the simulation evaluation, MCRP is tested in the real-world environment to evaluate the robustness of the multichannel protocol against unpredicted interference. The results proved that MCRP detects the interference-free channels and avoids channels with interference, which greatly reduces the effects of interference on the network. MCRP is compared against a single channel protocol.

Energy Efficient WSNs: MCRP helps to minimise the transmissions' energy consumption. This chapter presents the equation used to measure the energy consumption based on Contiki's existing energy estimation module that tracks the components' duty cycle. MCRP performance is compared to a single channel protocol's energy consumption in terms of the total energy over time, the energy per packet transmission and the energy of the forwarding packet. The results showed less energy consumption in MCRP due to its ability to detect and avoid communicating on the high interference channels.

Lifetime Energy Spanning Trees: A MCRP lifetime energy spanning tree is proposed to further improve the multichannel protocol by considering the energy level of each sensor. The equation and algorithm used to maximise the minimum energy tree are explained in this chapter. It aims to enable the network to be fully functional for a longer period of time by maximising the minimum sensor node energy level through topology reconstruction. The results showed an increase in the network lifetime for the improved tree compared to the initial tree.

To summarise, this thesis proposes a new multichannel protocol with a centralised controller. It presents a strategy for a better use of the spectrum while ensuring reliable communications by avoiding interference channels, which consequently improves the overall network energy consumption. The lifetime energy spanning trees strategy is developed to further prolong the network lifetime in addition to the multichannel protocol.

1.4 Thesis Outline

The remainder of the thesis is organised as follows. Chapter 2 introduces the state-of-the-art in the area of multichannel protocols. It also presents different approaches in the current research efforts towards energy efficient multichannel protocols. Chapter 3 presents the key features and mechanisms used in MCRP. It describes the proposed protocol's high-level design and strategies in the channel selection processes. It also presents the topology reconnection strategies that comply with MCRP design. Chapter 4 describes the implementation of the protocol in Contiki. The experimental results of the protocol are discussed and evaluated in Chapter 5 and Chapter 6. Chapter 7 describes the network's energy consumption. Chapter 8 introduces a new strategy to maximise the nodes' energy through lifetime energy spanning tree reconstruction. Chapter 9 concludes this thesis and discusses potential directions for future work.

Chapter 2

WSNs: Applications, MAC and Routing Protocols

2.1 Wireless Sensor Networks

A WSN is a network of sensor nodes that communicate using radio signals to collect data from the target area. These data that the sensor nodes send can be from sensor measurements, such as the temperature and movement in the specific area where the sensor nodes are located. The sensor nodes can be used for continuous sensing, event detection, location sensing and local control of actuators to control different components in the sensing device, such as adjusting the sensor parameters or moving the sensor node if it is a mobile sensor.

This chapter describes the available WSN applications, challenges and known issues that occur in WSNs. Many previous studies were done in order to maximise the lifetime of sensor networks while keeping the energy consumption at a minimum. This chapter also briefly describes the existing solutions for energy efficient multichannel protocols at the MAC and network layers, which prompted the work of MCRP.

2.1.1 Overview of Applications

There are five types of deployed WSNs that are commonly studied: terrestrial WSNs, underground WSNs, underwater WSNs, multimedia WSNs and mobile WSNs, which cover different types of environment, to deploy on land, underground, in water and in the air [127]. Unlike other sensor nodes, multimedia WSNs have the ability to monitor and track events in the form of video and audio, as they are equipped with cameras and microphones for multi-media data, which can enhance the existing WSN applications [4]. Mobile WSNs, on the other hand, can be any type of sensor nodes that have the capability to reposition and organise themselves in the network. Multimedia WSNs and mobile WSNs can be deployed on land, underground and in water depending on the requirement, therefore these are separated into different types, in addition to the main four main types of environment. However, multimedia and mobile WSNs are usually part of the terrestrial WSNs, owing to the tight network connectivity requirement between nodes, high bandwidth demand and energy consumption in multimedia WSNs.

The evolution of WSNs is driven by a number of emerging applications that focus on the importance of wireless sensors in applications such as smart grid, areas in smart cities, and automated home, building and industrial applications [62]. Smart grids could save considerable amounts of energy by improving the existing electrical grid power. Smart cities applications such as automated pollution monitoring and automated energy control in temperature and lighting can improve the environment quality, as the automation helps to increase energy saving in populated cities. In a smaller scale network, smart homes that are equipped with connected smart devices such as thermostats enable the user to control the smart applications remotely through a smartphone in addition to the devices' ability to work independently without human intervention, such as adjusting the thermostat temperature based on the daily weather forecast.

The applications of WSNs are important as sensor nodes can easily be deployed in all types of environment, installed and require minimal maintenance for a period of time. The main challenges in these applications are in terms of reliable event detection, securing high data rates for efficient data routing and the deployment dense or sparse nodes. The applications of WSNs can be categorised into five main monitoring and tracking applications, which are the environmental applications, health applications, home applications, military applications and other commercial applications [3]. These applications are briefly described in the next section, with examples for each category.

2.1.1.1 Environmental Applications

The environmental applications can be divided into two types: tracking and monitoring. The tracking applications are used to record the movements of animals such as birds, insects and small animals. Monitoring applications are used to monitor the environmental conditions, such as forest fire detection, flood detection, biocomplexity mapping [19], precision agriculture monitoring and volcanic monitoring [120].

In forest fire detection, the sensor nodes are used to relay the exact origin of the fire to the end users, to prevent it from spreading. ALERT [84] is an example of a flood detection system that is deployed in the United States. ALERT consists of several types of sensors such as rainfall, water level and weather sensors. In agriculture, the sensor nodes are used to monitor the level of pesticides in drinking water, soil erosion and air pollution in real-time. In volcanic monitoring, the sensor nodes allow measurements to be taken from locations that are otherwise inaccessible.

2.1.1.2 Health Applications

Sensor networks in health applications can be used to monitor human physiological data, such as detecting elderly people's behaviour in case of a fall, drug administration [83] in hospitals to minimise incorrect prescription of medication to patients, and to monitor and track doctors' and patients' locations in a hospital. Examples of these are telecare and telehealth [9].

Telecare is a system of wireless sensors that are placed around the house and can be a personal alarm in the form of a small wristband or pendant. A few examples of the system are a motion sensor that turns on the lights at night when someone gets out of bed, a pressure mat on the mattress to sense if someone gets back to bed or a sensor on the door in case it is not closed. If a risk is detected, it sends the alert immediately to a telecare monitoring centre.

Telehealth is a small piece of equipment to monitor health from home. It can be used to measure blood pressure, blood glucose levels, oxygen levels, weight or temperature. The measurements are automatically transmitted to a monitoring centre. The healthcare professional will be contacted if the information raises an alarm for actions to be taken.

2.1.1.3 Home Applications

In home automation [87], the smart sensor nodes and actuators can be built into appliances such as vacuum cleaners, microwave ovens and refrigerators, which allows them to form an interaction through the Internet. Two recent examples of home automation are Samsung SmartThings [93] and the Nest Thermostat [64].

Samsung SmartThings allows devices at home to be monitored and controlled from a mobile phone, such as thermostats and lighting. The Nest Thermostat is a self-learning thermostat that consists of activity sensors, temperature sensors, a humidity sensor and a Wi-Fi radio. These sensors allow Nest to learn the heating and cooling habits, which allow it to shut down due to inactivity to conserve energy. Nest is weather aware; it uses its Wi-Fi connection to get the weather condition and forecasts, and integrate the information to understand the effects of the outside temperature to the energy usage. Nest is also able to connect with other appliances that are Nest supported. The appliances can automatically start without any need to program it, as it learns from the other devices.

2.1.1.4 Military Applications

Examples of WSNs used in military applications include those to monitor friendly forces, equipments and ammunitions by attaching sensors, which report the status back to the base station; battlefield surveillance by covering critical terrains, routes, paths and straits with sensors and reconnaissance the opposing forces; assess battle damage, and to detect nuclear, biological and chemical attack by deploying sensors to explore areas and serve as warning systems to avoid casualties.

An example military application is PinPtr [100]. PinPtr is an experimental counter-sniper system. It is developed to detect and locate shooters by measuring the shot time arrival of the muzzle blasts and shock waves from the sensors that are densely deployed. The measurements are routed to the base station where the shooter's location is computed. PinPtr was demonstrated and evaluated in realistic urban environment from various US Army test facilities.

2.1.1.5 Other Commercial Applications

Other available commercial applications of WSNs are environmental control in office buildings such as controlling the air flow and temperature for different parts of the building, car thefts monitoring and detection within a specific region, inventory control management to track and locate the inventories in the warehouses, machine diagnosis in order to predict equipment failure for maintenance through vibration signatures gathered by sensors [63], and vehicle tracking and detection for parking purposes such as the Smart Parking from Streetline [108] and SmartPark [77].

Smart Parking solution is used in more than 40 cities and universities in North America and Europe. The system could make intelligent decisions using the data from the real-time and historical analytical reports to improve the parking ecosystem. The system detects vehicle occupancy in real-time, which simplifies the parking experience by guiding drivers to the available spaces. It can also guide officers to unpaid violations and overstays as the arrival and departure times are recorded, and to detect if a car is parked over the no parking and restricted zones. Similarly, SmartPark is another existing parking solution in the UK, currently operating in Birmingham and in the central London Borough of Westminster.

These applications enable drivers to find vacant space within the busy town and city centres more quickly.

2.2 Challenges and Issues

WSNs are widely used in various kinds of applications. This is because sensor nodes can be densely deployed, easy to install and require minimal maintenance over a period of time.

However, sensor nodes have *limited memory* capabilities. This only allows limited computational processes to be performed. It also suffers from *limited energy* capacities as the nodes are battery-powered and they will become faulty and not able to function once a certain threshold of energy level is reached. It also operates in an *unreliable radio environment* that is noisy and error prone, which *drains the sensor nodes batteries* at a higher rate.

These constraints have a major impact on the sensor nodes performance. In order to prolong the sensor nodes' lifetime thus, the network lifetime, the sensor nodes need to be able to cope with the limitations and be as energy-efficient as possible to guarantee good overall performance.

2.3 Maximising Lifetime and Minimising Energy

In WSNs, it is necessary to estimate the nodes' power consumption before they are deployed to enable accurate forecast of the energy consumption. The estimations are used to determine the nodes' lifetime before maintenance and batteries replacements are required, in order to have a functional network. Unfortunately, the node's lifetime is very dependent on the radio environment that can be unstable, noisy and error prone, which makes energy consumption to vary [61]. The network lifetime, however, depends on various factors such as the network architecture and protocols, channel characteristics, energy consumption model and the network lifetime definition. In order to increase the network lifetime, these information regarding the channel and residual energy of the sensors should be exploited.

There are various definitions of network lifetime that have been used. These definitions are application-specific as some applications might tolerate a considerable number of loss nodes, while some applications require a higher number of nodes, which any loss is considered critical to the network, such as in sparsely deployed nodes of an area. The definitions impact the performance differently, depending on the applications. The various definitions are:

- **The first node to die** - The network lifetime is defined as the first node to fail in the network [22, 130]. In [122], the simulation ends when a node reaches the energy level of zero.

- **The number of alive nodes** - The network lifetime is the number of remaining nodes as a function of time. The network has a longer lifetime with a higher number of remaining nodes [46, 66].

- **The number of nodes still connected to the sink** - The network is alive based on the remaining number of nodes to have coverage to connect to the sink [66].

- **The fraction of alive nodes** - The network lifetime is defined as lasting until the point where the proportion of surviving nodes drops below a threshold.

- **Packet delivery ratio** - The network lifetime ends when the packet delivery ratio drops below a preset threshold. GAF [126] uses this definition, where it has a constant traffic at all time. The network lifetime ends when the packet delivery ratio starts to drop.

In this thesis, the network lifetime refers to the first node to run out of energy. This definition is used because the failing node could be the main node that is connecting the other nodes to the sink. The network becomes non-functional even though the other nodes are alive, as the path to reach the sink is no longer exist. One of the aims of the WSN design is to extend the network lifetime under the given energy and node constraints without jeopardizing reliability and communications efficiency of the network.

There are four ways that have been explored in many studies to maximise the network lifetime, which are by introducing (i) energy efficient MAC protocols, (ii) energy efficient routing protocols, (iii) controlling the transmission power and (iv) using energy harvesting. These options are described in detail in the next few sections, introducing the differences and advantages, and the existing proposed solutions.

2.3.1 MAC Protocols

Many energy efficient MAC protocols have been proposed to prolong the network lifetime. The radio module that is controlled by the MAC protocol is the major energy consumer in WSNs. The radio uses nearly the same energy in all active operation modes such as the transmission, reception and idle modes [61]. During idle mode, the node is still awake even though it does not transmit or receive packets. Thus, it is important to reduce the radio usage to conserve the node's energy.

The main causes of energy consumption are nodes collision, overhearing and idle listening [8, 109]. Collision happens when nodes that are within each others' transmission range transmit simultaneously. The energy used in the collided transmissions is wasted as none of the nodes would receive the transmitted packet. A multichannel protocol is one of the solutions to overcome collision. Overhearing happens when a node receives irrelevant packets or signals that are not intended for the node. As the radio uses nearly the same energy in all operations, this drains the node's energy unnecessarily. In idle listening, the node keeps its radio on while listening to the channel for potential packets. The node does not know when it will be the receiver of the packet. A considerable amount of energy is wasted as the node keeps its radio on for a longer period listening to an idle channel when it does not receive or transmit packets.

A vast number of energy efficient MAC protocols have been developed to overcome these problems through duty cycling. Duty cycled MAC protocols allow the node to periodically alter the sleep state and listen state. By lowering the duty cycle, the node sleeps for a longer period instead of being permanently active. However, the node needs to have frequent check interval to avoid deafness problem while keeping overhearing to a minimum. This reduces the energy consumed by idle listening and overhearing. Many MAC protocols such as Energy-Efficient Multichannel MAC protocol (Y-MAC) [60] use duty cycle as the indicator to evaluate the energy efficiency performance.

2.3.2 Routing Protocols

Various energy efficient routing protocols for WSNs have been proposed and developed to ensure efficient packet delivery to the destination. The strategies that are used in routing protocols should ensure minimal energy consumption in order to prolong the lifetime of the network.

A major issue in WSN routing protocols is in finding and maintaining the optimal routes that are energy efficient. This is due to the energy constraints and unexpected changes in node status such as node failure or a node being unreachable. This causes the topology to be altered frequently to adapt to the changes. Rapid topology modification is important to avoid from having a disconnected network, which leads to a higher packet loss rate at the involved nodes if the routes are not quickly updated.

There are several routing techniques that have been studied, such as the flat, hierarchical and location-based routing protocols that are application dependent. These different routing techniques are explained in detail in Section 2.5.

2.3.3 Transmission Power Control

The term *topology control* has been used to mean two different things in WSN literature. Several authors defined topology control as the routing protocol techniques, to ensure that the nodes in the network remain connected and reachable throughout the network lifetime. Another definition of topology control is power control techniques, which act on the nodes' transmission power level [94]. The topology control term has been interchangeably used with power control. To avoid confusion, the term power control in this thesis refers to the techniques to control the transmission power levels (which then affects topology).

In power control, a node has control over the transmission range of the node's radio, which can be manipulated to benefit the network. The power adjustment approach allows the node to vary the transmission power to form a connected network that minimises the energy incurred in transmission. The nodes collaboratively adjust the power value to find the appropriate transmission power, which enables the nodes to transmit at a lower transmission power than at the maximum. However, a sparse network would require a higher transmission power than a dense network to be able to transmit to the nearest node.

The power control technique eliminates the redundant links that are not used for transmissions by fixing the area of coverage, thus routing. This reduces collisions as the nodes have lesser links within the smaller area. However, the nodes need to change the transmission power to adapt to the coverage changes if the nodes need to cover a larger (increase transmission power) or smaller (decrease transmission power) area.

As the transmission ranges are relatively short, the nodes can simultaneously transmit packet without interfering each other, thus reducing congestion from retransmissions. Although power control improves the network traffic flows, it does not reduce the nodes' power consumption as it depends on the radio duty cycle. The radio duty cycle controls the nodes' sleep-awake periods, which consume power during the awake period regardless of the transmission powers. Power savings due to transmission powers are therefore negligible [8]. The authors in [97] suggested multichannel protocol combined with transmission power control to be a promising strategy for energy efficient and reliable network based on the observations in their studies.

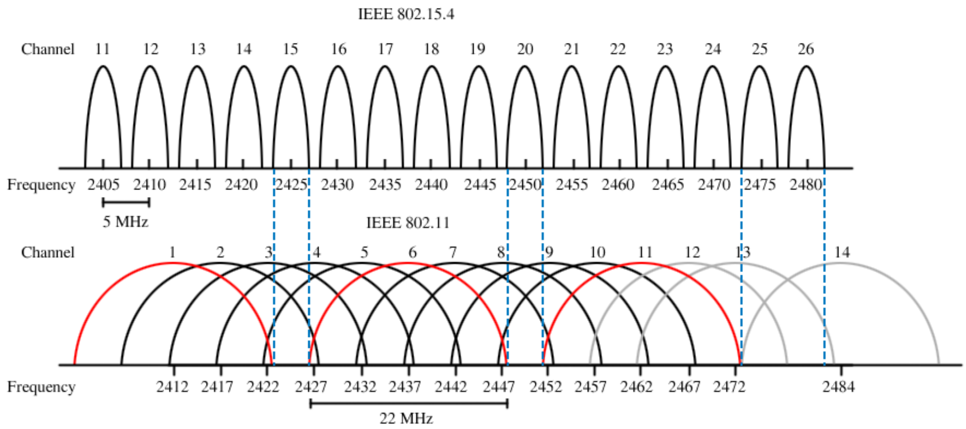


Figure 2.1: IEEE 802.15.4 and IEEE 802.11 frequency channels in the 2.4 GHz ISM band

2.3.4 Energy Harvesting

Energy harvesting is when a node tries to replenish its energy by using other energy sources such as solar cells [96, 118], vibration [42], fuel cells, acoustic noise and a mobile supplier [127]. The use of solar cells is to harvest energy from light [52]. There is also work in using robots as mobile energy suppliers to deliver energy to the nodes [99]. This allows a longer network lifetime as the node has restored its energy. However, energy harvesting depends on various environment factors such as light, vibration and heat to be generated and converted to the usable electrical energy.

2.4 Multichannel MAC Protocols

In single channel MAC protocols, nodes are configured to use a single channel throughout the nodes' lifetime. Frequency agile MAC protocols, on the other hand, allow the nodes to switch to different channels during run time. This is possible, as recent radio chips take less than 100μs to switch to a different channel. The channel switching delay is negligible in the WSN context where the packet rates are low. This makes multichannel protocol attractive for use in WSNs. A multichannel protocol has the advantage of an increase in robustness against external and intra nodes interference, which as a result, improves the network traffic flow.

2.4.1 Introduction

There have been many proposals in multichannel communication, which use the duty cycling technique to alter the node's sleep and listen states. The duty cycle is an important mechanism that helps reducing the node's energy consumption by turning the radio off (sleep) when the node is not sending or receiving packets. However, adjusting the duty cycle does not solve the interference problem as external interference is unpredictable, especially with Wi-Fi interference as it can potentially collide with four IEEE 802.15.4 channels, as each Wi-Fi channel is 22 MHz wide while IEEE 802.15.4 is 5 MHz wide. Figure 2.1 shows the IEEE 802.11 and IEEE 802.15.4 frequency channels in the 2.4 GHz ISM band. There are only a few channels that do not overlap with Wi-Fi, which are channel 15, 20, 25 and 26. However, avoiding all channels with Wi-Fi would overload the channels that 802.15.4 uses for transmissions. Therefore, the overlapping frequencies are used by introducing other means to improve the technologies coexistence [68].

A multichannel protocol is a preferable solution to improve resilience against interference and maintain reliable communications. However, not all channels are free from interference, thus, there is a gain to hop to another channel when the quality of the channel deteriorates. The authors in [97] found that the channel reliability changes over time in non-cyclic manner, thus no specific channels could achieve a long term reliability. Infrequent channel hopping is required to ensure network connectivity.

Two commonly used types of channel hopping [119] are blind channel hopping and whitelisting. In blind channel hopping, the nodes choose a channel from all available channels. Whitelisting, on the other hand, gives a set list of channels that avoids those that are known to commonly suffer interference.

Existing duty cycle multichannel MAC protocols can be categorised into two types: synchronous and asynchronous systems. These are also referred to as reservation-based protocols and contention-based protocols by some authors. A synchronous system is a system that requires a tight time synchronisation between the nodes. It uses time-scheduled communication, where the network clock needs to be periodically synchronised to compensate for time synchronisation error in order for the nodes not to drift in time [60]. The system requires dependency on the time synchronisation and network topology. The knowledge of the network topology is required to be able to establish a schedule for the nodes to access the channel to communicate with the other nodes.

An asynchronous system, on the other hand, does not require synchronisation and topology knowledge, but instead it is a sender or receiver initiated communication. The nodes compete to access the channel to transmit, such that the node postpones its transmission if it senses that the channel is busy, by sending preamble packets, to avoid interfering with the current transmission. In asynchronous systems, the nodes are able to self-configure without time synchronisation and this can have advantages. There are many studies done in multichannel protocol in both categories.

Multichannel communications have potential benefits in wireless networks that include improved resilience against external interference, reduced latency, enhanced reception rate and increased throughput. A set of existing multichannel MAC protocols are reviewed and compared in the next section, highlighting their features and limitations.

2.4.2 Synchronous Systems

In synchronous systems, the multichannel MAC protocols employ Time Division Multiple Access (TDMA). It allows the channels to be divided into different time slot. TDMA-based MAC protocols allocate time slots to the nodes for data transmission or reception [60]. This helps to avoid collision between the nodes during transmission, as the nodes have their own time slot. However, it has a higher latency as the node has to wait for its assigned slot before it is able to transmit a packet.

Timeslotted Channel Hopping (TSCH) [111], Orchestra [38], Multi-Channel Lightweight Medium Access Control (MC-LMAC) [51] and YMAC [60] are a few examples of the existing synchronous systems. These multichannel MAC protocols are selected for review.

2.4.2.1 TSCH

The Timeslotted Channel Hopping (TSCH) [111] is a MAC protocol that uses time synchronisation and channel hopping to increase reliability in the network. The nodes in TSCH are fully synchronised. The nodes are assumed to be equipped with clocks as the nodes need to maintain tight synchronisation. The clocks in different nodes could drift in time, therefore the nodes need to periodically resynchronise their clocks with the time-source neighbour in the absence of data to transmit. The nodes also provide their time during synchronisation to the neighbours. When the nodes have data to send, the timing information is added to the packet, which simplifies the synchronisation process as the nodes are resynchronised each time they exchange data.

It is designed for optimisation, customisation and it simplifies the process of merging TSCH with protocol stacks based on IPv6, IPv6 over Low Power Wireless Personal Area Network (6LoWPAN) and Routing Protocol for Low Power and Lossy Network (RPL). TSCH defines the mechanism to set up the schedule and control the resources allocation to each link in the network topology for execution. It also defines the mechanism that signals when a node cannot accept an incoming packet. However, it does not define when the node should stop accepting packets.

In TSCH, time is sliced up into time slots that are appropriate for the traffic flow size. The time slot is set to be long enough to enable the sender node to send a maximum size of MAC frame to the receiver node and for the receiver to send an Acknowledgement packet (ACK) frame to notify the sender that the frame has been successfully received.

Slotframes contain a group of time slots of equal length and priority, where the slotframe repeats continuously over time. The size of the slotframe depends on the application implementation. Shorter slotframe has the advantage of more available bandwidth as the result of frequent repetition of the same time slot, but at the cost of higher power consumption.

A single element in TSCH schedule is called as a *cell*. The cell can instruct the node to transmit, receive or sleep. In a transmit cell, the outgoing buffer is checked for a packet that matches the scheduled neighbour for that time slot. Similarly, in a receive cell, the node listens during the reserve cell for possible incoming packets. The cell can also be marked as both transmitting and receiving. However, transmission takes precedence over reception. TSCH schedule also indicates the channel and address of the node for communication. The channel in TSCH is referred to as *channelOffset*, which is the row in TSCH slotframe. Each scheduled cell is dedicated for the node. However, a cell can be shared where multiple nodes can transmit on the same frequency at the same time. TSCH defines a backoff algorithm to avoid transmissions from nodes in the shared cells from congesting the network.

Absolute Slot Number (ASN) is a timeslot counter in TSCH that calculates the communication frequency for the sender and receiver nodes. ASN value is computed using Equation 2.1.

$$ASN = cycle * slotFrameSize + slotOffset \tag{2.1}$$

The calculation from ASN and *channelOffset* is translated into a different frequency at different slotframe cycles using Equation 2.2 where the *channelOffset* represents the 16 frequencies available, and the *numberOfFrequency* is the total number of frequencies (16 channels).

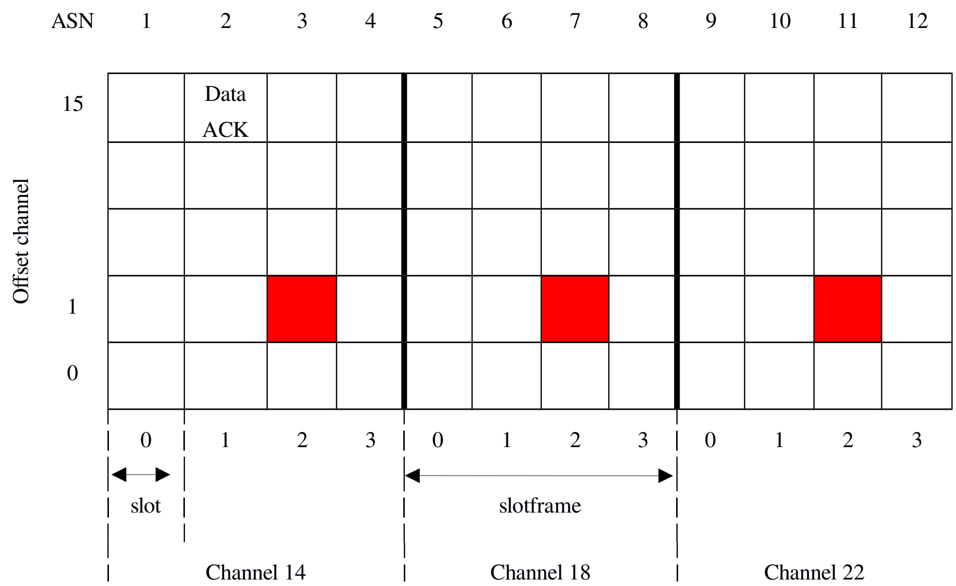


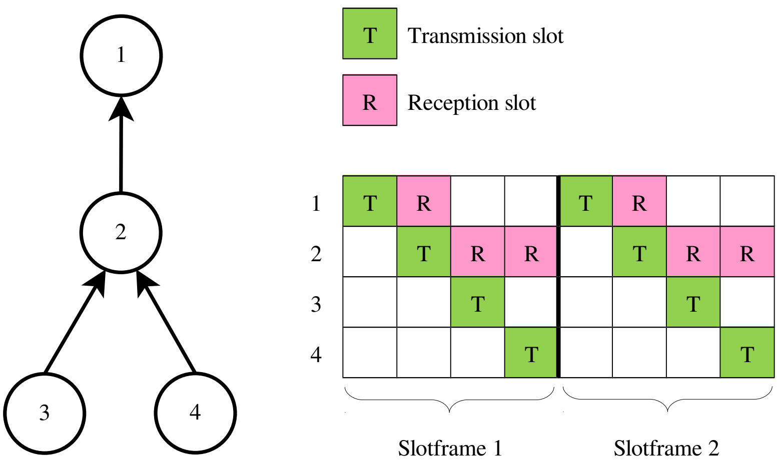
Figure 2.2: TSCH schedule

$$frequency = F(ASN + channelOffset) \mod numberOfFrequency \tag{2.2}$$

The ASN value changes at the next iteration, which results in a different frequency computed for the cycle. This results in channel hopping where the pairs of neighbours hop between different channels at each iteration.

Figure 2.2 shows an example of TSCH schedule. The red box represents the node's time slot in three cycles. During the first cycle, using Equation 2.2, it resulted in the value of 4, which is the channel 14. The 16 available channels are in the range of 11 to 26 where the fourth channel represents channel 14. For the next cycle, the ASN counter has a different value, which gives a different frequency, which is channel 18 and channel 22 for the next iteration.

The advantage of channel hopping is to have retransmission on a different channel than it was transmitted previously. It hops to another channel on the next cycle. This increases the likelihood of succeeding compared to retransmitting on the same channel, thus, forming a more stable topology. Nodes on different channels are allowed to run simultaneously in the same time slot because the nodes do not interfere with each other's transmissions. The channel hopping technique helps to combat external interference.



(a) Network topology (b) Slot coordination

Figure 2.3: Orchestra default sender-based shared slot

2.4.2.2 Orchestra

Orchestra is a timeslot protocol that is based on TSCH [111] for low-power communication. TSCH uses channel hopping, which uses a different frequency in each slotframe of a node's timeslot. This enables packets to be communicated on different frequencies to increase the probability of succeeding if the transmission fails in the previous frequency.

Orchestra uses TSCH and Routing Protocol for Low Power and Lossy Network (RPL) [121] routing protocol local state of the neighbours and parents to maintain the schedules autonomously while benefiting from TSCH robustness. Orchestra has a flexible schedule, which allows the schedule slots for different traffic to be automatically installed or removed as the RPL topology evolves. Orchestra does not introduce any scheduling traffic overhead as it does not require a centralised scheduler, negotiation or path reservation.

Orchestra identified four main types of slots, which are the (a) common shared slots where all nodes shared the slot in both transmission and reception, such as in running the RPL protocol as the nodes wakeup simultaneously, (b) receiver-based shared slots where the receiving slot is fixed for every node, (c) sender-based shared slots where the sending slot is coordinated and (d) sender-based dedicated slots where contention-free communication is guaranteed. Orchestra uses standard TSCH exponential backoff to resolve contentions that arise except in the sender-based dedicated slots, where the lost packets are retransmitted on the next slot.

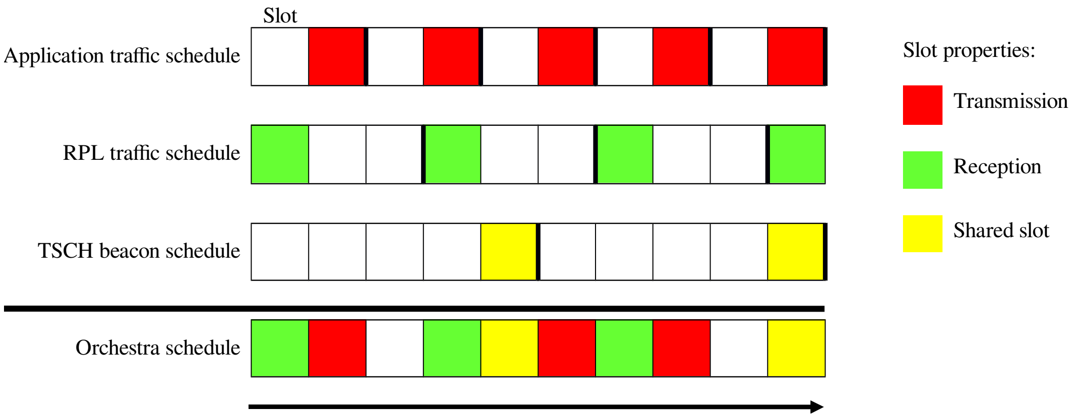


Figure 2.4: Orchestra schedule slots

By default, Orchestra uses the sender-based shared slots. Figure 2.3 shows an example of Orchestra sender-based shared slot with channel hopping. Figure 2.3(a) shows an example of a simple topology and Figure 2.3(b) shows the slot coordination based on the example topology. Slotframe 1 is on a different channel than in Slotframe 2. Each node has its individual transmission slot, which is repeated at every cycle (next slotframe). In the example, as this is a sender-based slot, node 2 requires two reception slots for each route from node 3 and node 4.

In Orchestra scheduling, the slotframes are each dedicated to a particular type of traffic plane of different lengths. These slotframes are the MAC or TSCH beacon schedule, the RPL signalling traffic schedule and the application data schedule. These different types of traffic have different priorities, which as a result, reduce contention.

Figure 2.4 shows an example of the schedules in Orchestra allocated to specific traffic planes that have different lengths. In Orchestra, TSCH beacon takes precedence over RPL, and RPL takes precedence over other application traffic if the slot overlaps.

Orchestra is an autonomous scheduling of TSCH in RPL networks that benefits from TSCH channel hopping and it is able to adapt the RPL topology changes to the schedule slot for successful communications.

2.4.2.3 MC-LMAC

Multi-Channel Lightweight Medium Access Control (MC-LMAC) is a fully distributed schedule-based multichannel MAC protocol that is based on a single channel protocol Lightweight Medium Access Control (LMAC) [116]. The nodes periodically use a timeslot to schedule the transmission to avoid contention. MC-LMAC does not require a centralised

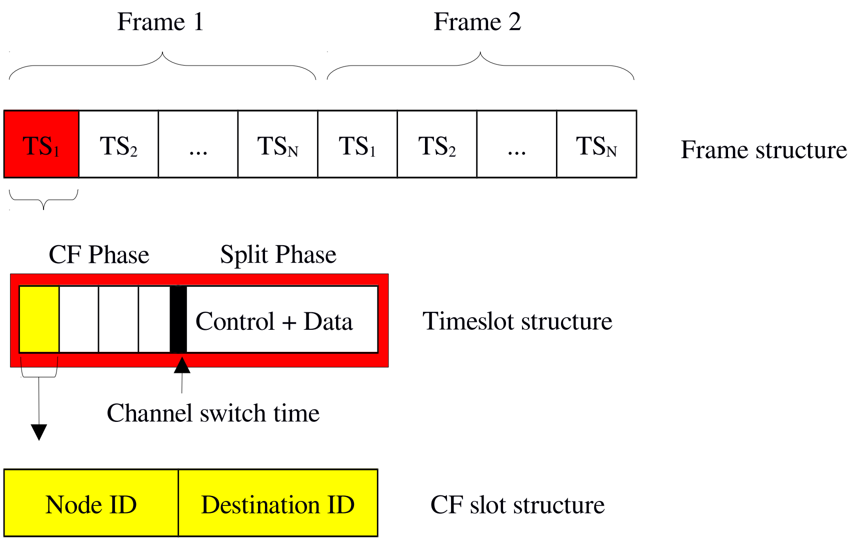
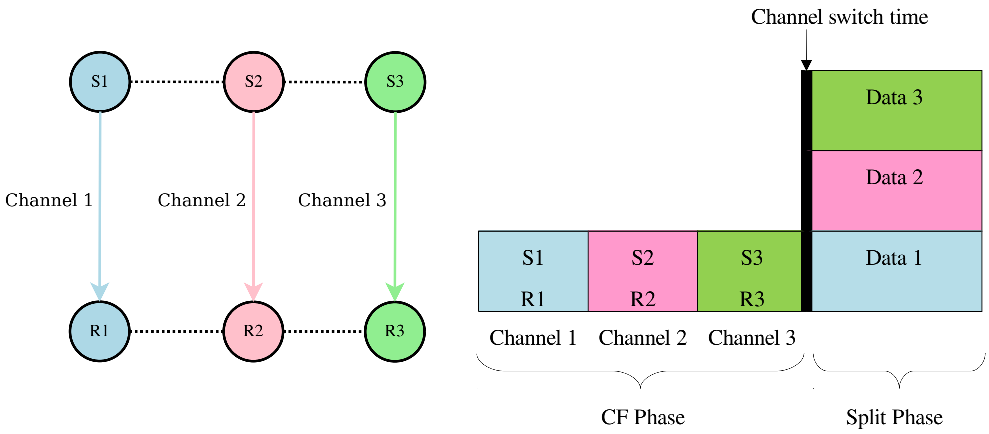


Figure 2.5: MC-LMAC scheduling

scheduler for timeslot allocations, instead, it is done locally by the nodes by exchanging information of their slots and channels with the local neighbours.

In MC-LMAC, the timeslots are selected with channels. The node can use the same timeslot used by a two-hops neighbour, if it is on a different frequency. However the node cannot use the timeslots on any frequencies that are used by the neighbours. The timeslot is selected autonomously. The node uses the same timeslot in the next frame if it does not conflict with the other nodes' transmission in that slot. Otherwise, a new time slot is selected. The timeslot list is called *occupied slot vector* where it stores the information about the neighbours' occupied slots. The slot vector is per channel, where the node can selects a timeslot for each channel given that the timeslot is free. The occupied slot vector is transmitted during the node's timeslot to the potential transmitters. All nodes are given the opportunity to select an empty slot for transmission.

A Time slot (TS) consists of a Common frequency phase (CF) and a *split phase* as shown in Figure 2.5. The number of timeslots per frame is *N*, $(TS_1, \ldots\ TS_N)$. All nodes listen on the common channel at the beginning of each timeslot in the CF phase to exchange control information with the neighbours. The common channel can be used for data transmission. The control information in the CF phase consists of the node's id and the intended destination id. The receivers listen during the whole CF phase. If it is the intended destination, the node switches to the sender's channel during the *channel switch time* before



(a) Network topology (b) MC-LMAC scheduling

Figure 2.6: MC-LMAC protocol

proceeding to send the packet in the split phase, otherwise it goes into the passive state. MC-LMAC uses the CF slot number as the senders channel number to avoid sending an extra transmission to the destination node.

Nodes can send broadcast messages by transmitting a broadcast address during the CF slot, where the receivers switch to the sender's channel. A dedicated broadcast channel is not required. However, the CF duration increases when more channels are used, which results in longer listening period thus energy to wait for potential incoming packets.

The senders and the intended receivers switch to the channel where the control message and data transmission will take place in the split phase. The sender sends a control message in the form of preamble packets before proceeding with transmitting the data message. The control message that is transmitted in the split phase includes the occupied slots list. The node also sends the current slot and slot numbers in the control message prior to data transmission to detect synchronisation error by comparing the slot and frame numbers that it receives in the control message with its slot and slot number.

Figure 2.6 shows an example of MC-LMAC protocol with 3 channels. Figure 2.6(a) is the example topology, which corresponds to the scheduling shown in Figure 2.6(b). The CF phase has the sender and receiver ID; S1 to send to R1 on channel 1, S2 to R2 on channel 2 and S3 to R3 on channel 3. The receivers switch to their senders channel during the channel switch time before the senders proceed to send the data on their own channels. In MC-LMAC, the synchronisation is done by synchronising nodes near to the sink with the sink, and continues hop by hop, where the nodes synchronise with the parents.

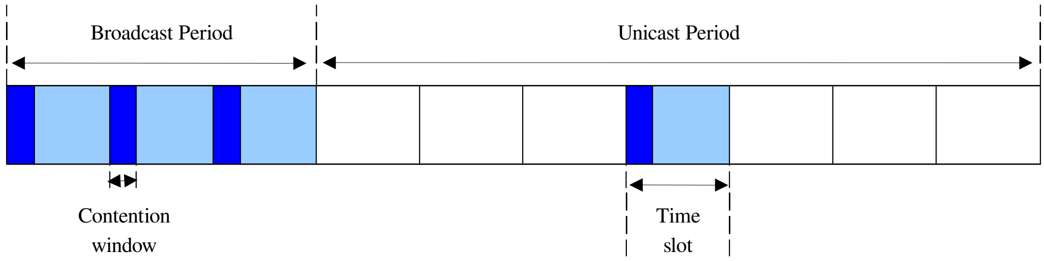


Figure 2.7: Y-MAC scheduling

2.4.2.4 Y-MAC

Y-MAC [60] proposed a multichannel MAC protocol that uses a light weight channel hopping mechanism. In Y-MAC, time is divided into several fixed-length frames. Each frame consists of a broadcast and unicast period. The broadcast traffic is separated from the unicast traffic for a more reliable broadcast, where they do not share the same queue. Figure 2.7 shows Y-MAC scheduling. At the start of the broadcast period, all nodes must wake up to exchange broadcast messages. The nodes switch to the base channel to transmit or receive the broadcast message. Broadcast messages are only exchanged during the broadcast period. The nodes turn the radio off if there is no incoming broadcast message. The nodes will wake up again during the unicast traffic time slot. Y-MAC exploits multichannel for unicast to reduce the packet delivery latency while using a single channel, which is the base channel for broadcast messages.

Y-MAC is a receiver based scheduling where the node checks the channel for incoming packets in its receive time slot. The time slot length is defined to be long enough to receive one message. The potential senders have to compete to be able to transmit. However, only the contention winner can transmit the packet to the receiver. The sender node sends a preamble until the end of the contention window if the channel is clear to withhold competing transmissions. At the end of the contention window, the receiver wakes up to receive the data.

Figure 2.8 shows an example of Y-MAC channel hopping. The receiver initially starts the hopping sequence on the base channel to receive the data. The receiver and potential senders hop to another available channel according to the hopping sequence to receive the following packet. The potential senders that have pending messages for the receiver will hop to the next channel and compete to transmit. The burst of messages ripple across channels, which means that only one node uses the base channel at a time. This guarantees

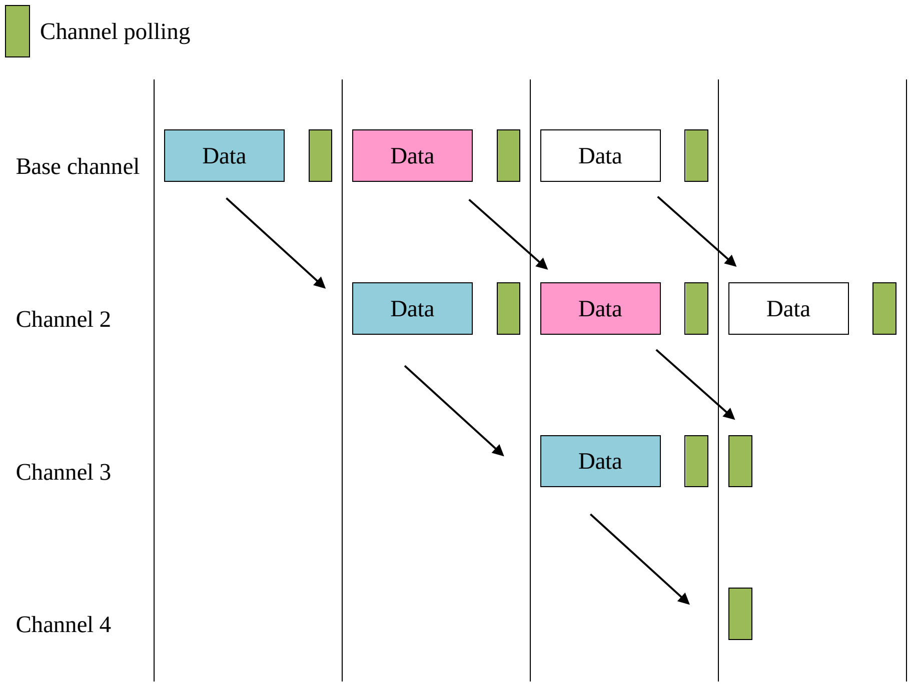


Figure 2.8: Y-MAC channel hopping

that each node will receive a packet on the base channel before it hops to another channel. The hopping sequence ensures that for any particular channel, there is only one node among one hop neighbour. The receiver node transmits a small and independent packet at the start of the time slot to notify the potential senders if it waits in the next time slot for the senders to retry on the next channel (channel polling).

In Y-MAC, the nodes exchange the time remaining to the start of the next frame period. This information is included in the control message that is sent periodically as a broadcast to minimise the control overhead for time synchronisation. The receiving node that receives the time synchronisation information adjusts the expiration time of its timer by averaging the time remaining to compensate any timing error so that the start time to the next frame period is shorter. Time synchronisation is an important aspect in ensuring the network connectivity for communications.

2.4.3 Asynchronous Systems

Recent asynchronous multichannel MAC protocols are Chrysso and Multichannel Contiki-MAC protocol (MiCMAC). These protocols are using the Contiki operating system. MiC-MAC is built based on Contiki MAC protocol (ContikiMAC), the default radio duty cycling in Contiki 2.7 that works in a single channel protocol. The details of these are explained below. The single channel ContikiMAC is also explained, as the duty cycling mechanism in ContikiMAC is important in MCRP, the protocol proposed in this thesis.

2.4.3.1 ContikiMAC

ContikiMAC [33] is the default radio duty cycling mechanism in Contiki 2.7. It is an asynchronous system where it does not need scheduling, signalling messages, and additional packet headers. It uses periodical wake up to listen to incoming packets from the neighbours. Periodical wake up has been used by many protocols, such as Berkeley Media Access Control (B-MAC) [89], Low Power MAC protocol (X-MAC) [17] and Physical and Link Layer Boundaries MAC protocol (BoX-MAC) [80]. ContikiMAC default wake up frequency is set to 8 Hz, which results in a wake up interval of 125 ms. Frequent wake up would enable quicker packet detection in the case of frequent packet transmissions at the cost of higher network power consumption.

The receiver is kept on when it detects a packet transmission during a wake up. ContikiMAC wake up uses an inexpensive Clear Channel Assessment (CCA) that relies on the threshold of the Received Signal Strength Indicator (RSSI) to signify the radio activity on the channel based on the signal strength measurement. A positive CCA represents a clear channel if the RSSI is below a threshold and the CCA returns a negative value if the channel is currently in use. The ContikiMAC CCAs do not detect packet transmission. They are used to detect the activities on the radio signal, which could be that (i) a neighbour is transmitting to a receiver, (ii) a neighbour is transmitting to other receivers, or (iii) other devices that radiate radio energy.

ContikiMAC uses a fast sleep optimisation to enable the receivers to quickly go to sleep in the case of spurious radio interference that is a false positive wake up. This allows run-time optimisation of the energy efficiency in transmissions. The receivers can go back to sleep if (i) the duration of the radio activity is longer than the maximum packet length, (ii) the silence period is longer than the interval between two successive transmissions or (iii) the start of packet is not detected.

Another feature of ContikiMAC is the transmission phase-lock mechanism. This feature has been suggested by Wireless Sensor MAC (WiseMAC) [39] previously and has been used by other protocols. The phase-lock optimisation manages a list of neighbours and their wake up phases.

When a sender has a packet to send during the transmission phase-lock, the sender

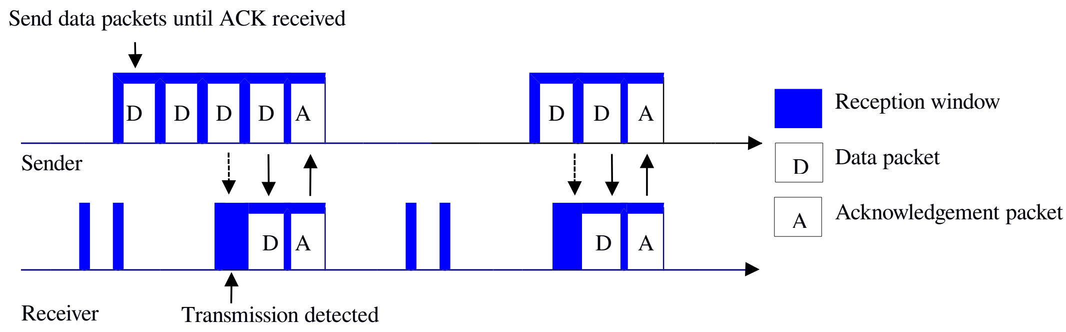


Figure 2.9: ContikiMAC unicast transmission with phase-lock

repeatedly sends the packet until it receives a link layer acknowledgement from the receiver to indicate that the packet has been successfully received. Figure 2.9 shows ContikiMAC unicast transmission phase-lock. A sender can learn the receiver's wake up phase through the link layer acknowledgement. This reduces the number of transmissions required significantly in the next transmission as the sender has learned the node's wake-up phase and can send the packet shortly before the receiver is expected to be awake, which as a result, reduces the network congestion.

However, a broadcast packet does not result in a link layer acknowledgement. The packet is instead repeatedly sent in the full wake up interval to reach all neighbours. During broadcast, the sender can turn the radio off between each packet transmission to save power as it does not expect to receive any link layer acknowledgement.

2.4.3.2 MiCMAC

MiCMAC [5] is a distributed channel hopping variant of ContikiMAC. It inherits ContikiMAC basic design and it is further extended to support multichannel. It also extends ContikiMAC's phase lock to include *channel lock* for wake up channel. MiCMAC is independent from the other layers in the protocol stack. It is compatible to run with the RPL routing.

In MiCMAC, the node switches to a different channel each time it wakes up. The channel is generated using a Linear Congruential Generator (LCG) for a pseudo-random sequence. The generated sequences are random and use each possible number within the range once before the sequence is repeated. MiCMAC uses a predefined hopping sequence that is provided in a static table of all sequences instead of generating the hopping sequences at runtime to increase optimisation. The sequence is selected according to the node's MAC address.

When communicating with a neighbour for the first time, the sender transmits strobes repeatedly on a channel for a maximum number of channels wake up. The sender picks any channel. The receiver wakes up on a different channel each time following the pseudo-random sequence and it will wake up exactly once on the sender's strobing channel. The receiver sends an ACK that includes the node's pseudo-random generator parameters. The sender will use this information and the number of periods elapsed since the last successful unicast with the receiver to generate the next wake up channel for the node.

The sender switches to the receiver's expected channel just before it wakes up, checks the radio activity using CCA and sends the packet if the channel is clear. The receiver will reply with an ACK to the sender. The sender updates its information of the receiver's wake up time and channel before it goes back to sleep. If the ACK is not received, the sender assumes that the receiver's wake up time and channel is wrong. The sender updates the receiver's information.

MiCMAC supports broadcast and introduces two variants of MiCMAC; MiCMAC and MiCMAC with broadcast support (MiCMAC-BC). In the basic MiCMAC broadcast, only one of the available channels in the sequence is strobed continuously for a maximum number of channels wake up. It has the disadvantage of increased energy and channel used from the broadcast message. MiCMAC-BC, on the other hand, uses a dedicated broadcast channel. The nodes wake up on the broadcast channel at every period in addition to the unicast pseudo-random wake up channel. However, it has the disadvantage of reduced robustness as the broadcast channel is fixed and MiCMAC requires two wake up at every period, which is for the broadcast, followed by the unicast channels.

Figure 2.10 shows MiCMAC-BC and unicast transmissions where *B* represents broadcast, *U* for unicast and *R* for received packets. The colour red, green and orange represent different channels where red is the broadcast channel. The broadcast channel is fixed, thus the nodes will always wake up on the same broadcast channel. Blue represents received packets on the channel that the node wakes up on. Each node has two wake up where the first wake up is on the node's channel (different at each wake up) and the second wake up is on the broadcast channel. The example shows the sender sending a broadcast message and followed by a unicast message. When the sender sends a broadcast message, the receivers will detect it during the second wake up. When the sender has a unicast message

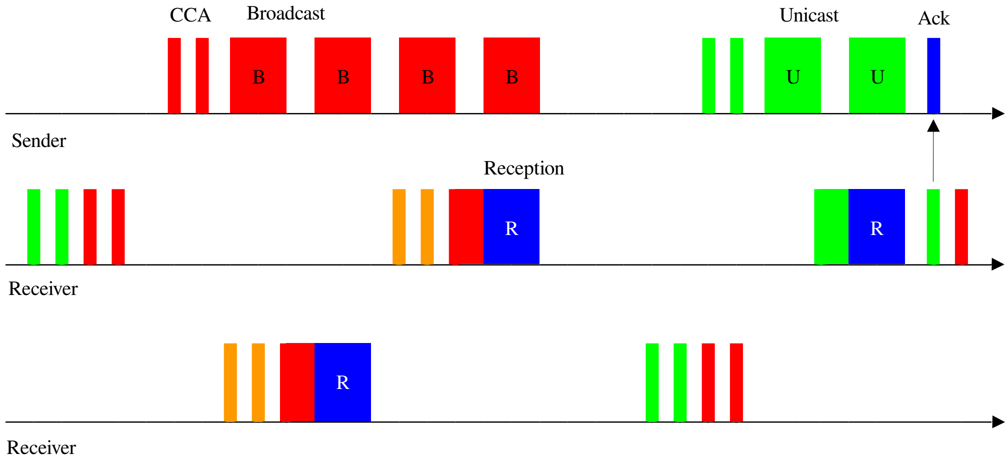


Figure 2.10: MiCMAC unicast and MiCMAC-BC transmissions

to a receiver, the sender and receiver should be on the same channel (green) to successfully receive the packet. The receiver sends an acknowledgement on the same channel before proceeding to the second wake up to check for any broadcast message.

In MiCMAC, the performance is degraded when it considers all 16 channels as it includes the high interference channels for transmissions. It also increases the broadcast and channel-lock costs. MiCMAC showed an improved performance with 4 channels.

2.4.3.3 Chrysso

Chrysso [56] is a multichannel protocol extension that is specifically tailored for data collection applications. Chrysso is implemented in Contiki 2.4 using *collect* routing protocol, which is a data collection. Chrysso switches the channel of the nodes that are affected by the external interference to a new set of channels when the interference is detected to evade the interference source.

Chrysso allows the parent node to coordinate the channel switch when interference is detected for each individual parent-children group. Chrysso operates on two channels as shown in Figure 2.11; one for incoming (called inchannel) and the other channel is for outgoing (outchannel) traffic. Chrysso maintains a pre-defined logical list of available channels. The parent and children use this list to ensure consistency when they are switching to the next channel. Chrysso implemented five channels, which are channel 26, 14, 20, 11 and 22 and evaluated Chrysso's performance using these channels in this particular order.

Chrysso consists of a set of control loops; the inner and outer loops, that decides to

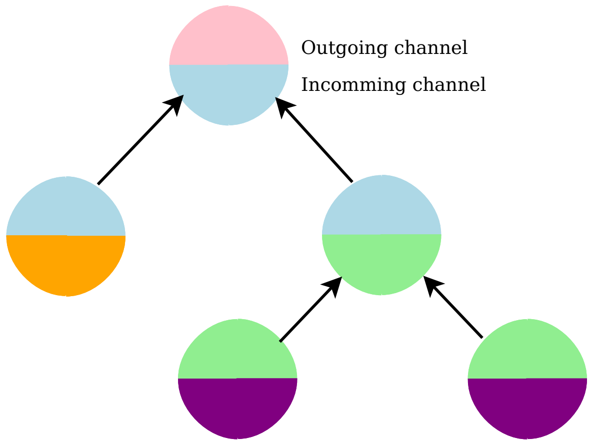


Figure 2.11: Chrysso

maintain or switch the parent-children's channel. The inner loop is responsible for the parent-children channel switching coordination when there is external interference. The child node collects data from the channel quality monitor periodically, which is then included onto the data packet. The parent uses the average over congestion backoffs values to measure and determine if there is external interference in the current measured channel quality. If the computed value exceeds a predetermined threshold, the children are notified of the channel switching request before the parent switches to the next inchannel.

Chrysso uses the outer loop during severe interference that blocks any communication where the parent-children channel switching coordination could not be invoked. The outer loop is a watchdog mechanism, where autonomous channel switching is initiated when the inner loop could not be triggered. The nodes, both parent and children independently switch to the next channel in outer loop. The child node monitors the number of failed transmissions. If it exceeds a predefined threshold, the outchannel is switches as it indicates that the channel has severe interference. Likewise, the parent records the number of packets it received and switches to the next inchannel instantly and autonomously if the received packets are below a pre-set threshold value of the expected packets.

The watchdog initiates the *scan mode* to find a new parent when the nodes have lost contact with the parent after the channel switches. The node scans through all available channels except the previously used outchannel to find a new parent as neighbours are now operating on different channels. The scan mode is triggered on demand as it uses additional overhead for processing and consumes energy.

Protocol	Medium Access	Channel Assignment	Channel Switching	Common Period	Broadcast
TSCH	TDMA + collision window	Dynamic	Once per cycle	No	Yes
Orchestra	TDMA + collision window	Dynamic	Once per cycle	No	Yes
MC-LMAC	TDMA	Senders	Once per time slot	Yes	Yes
Y-MAC	TDMA + collision window	Dynamic	Once per time slot	Yes	Yes
MiCMAC	MiCMAC	Dynamic	Once per wake up time	Yes	Yes
Chrysso	Contiki-MAC	Dynamic	When required	No	No

Table 2.1: Comparison of studied MAC protocols

2.4.4 Comparison and Discussion

The reviewed MAC protocols features are summarised in Table 2.1. There are several issues that need to be discussed.

1. **Synchronous versus asynchronous design** - Both designs have the advantages and disadvantages. However, synchronous MAC protocols require the network topology to be known before it can schedule timeslots to avoid conflict between the nodes. It can also be costly due to consuming and wasting bandwidth for synchronisation. Asynchronous MAC protocols depend on the channel condition and the nodes compete for the channel access.

2. **Sender versus receiver initiated design** - Most MAC protocols presented are both sender and receiver initiated where the channel decisions depended on both nodes.

3. **Channel hopping design** - In most of the MAC protocols studied, the protocols select several channels instead of considering all available channels. This limits the spectrum usage of the frequencies.

4. **Dedicated versus dynamic control channel** - Most MAC protocols use a common control channel to allow easy negotiation and synchronisation. As the common channel is fixed and known, it is vulnerable to attacks, which will affect the performance

of the whole network. Dynamic control channel can avoid from overloading specific channel by selecting different channels depending on the channel conditions. However, it requires other ways for the channels to be known, such as using a fixed set of channel hopping sequences.

5. **Broadcast support** - Most MAC protocols presented provide a broadcast support except for Chrysso. The protocol specified a broadcast period at every wake up on a fixed channel, which reduces the robustness of the protocol as broadcast will occur on the same channel each time. TSCH, on the other hand, treats broadcast message the same as a unicast message, where it needs to select a slot before it can proceed but with a broadcast MAC address as the destination.

Based on these reviews, the MAC protocols have many features that ease the multichannel processes. However, to further improve multichannel protocols, cross layers decisions are required, such as implementing routing protocol that takes into account the multichannel criteria (recalculate the route to use a different node with a different channel instead of fixing the route and switching the node to a different channel). This could lighten the processing load at the MAC protocol and to allow switching decisions to be determined at run time to compute better decision. MCRP is designed based on these findings.

The broadcast support in MAC protocols remains as an open question as it has both advantages and disadvantages. The broadcast message could be sent as a unicast message to reduce the number of transmissions instead of sending the same broadcast message over all available channels. MCRP is currently supporting broadcast message on a fixed channel.

Most MAC protocols use dynamic channel assignment, which means the channel is changed at every cycle (or when the current channel is not performing well) depending on the protocol. A list of selected channel hopping sequence is used. This has the disadvantage of having to check the channel conditions at every location before the channels are included in the list. It is possible to use blind channel selection. However it might not perform as well as when the channels are carefully selected. MCRP is designed to be able to decide on the channels at any location without the need to have any knowledge beforehand. This allows better use of the frequency spectrum.

The prior protocols avoid real-time channel selection because it requires higher computational power and memory to run the channel selection in real-time. MCRP overcomes this problem by using a centralised controller, which has no memory and energy limitations,

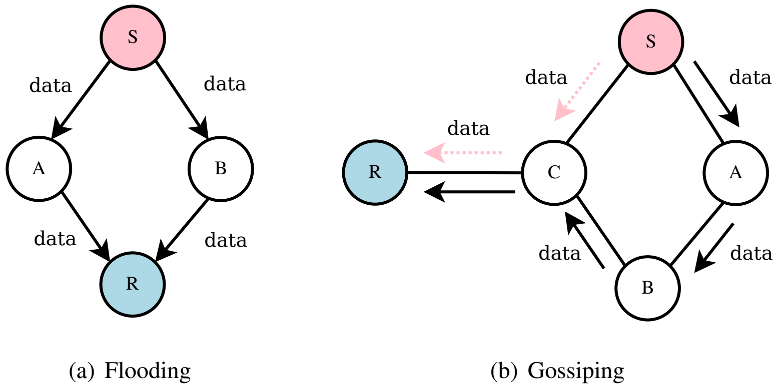


Figure 2.12: Classic data relay approaches

and it has the knowledge of the whole network, which allows it to make the channels and routing decisions (for lifetime energy spanning tree) as necessary.

2.5 Routing Protocols

Routing protocols are responsible in routing data from the sender to the intended receiver across the network. In WSNs, the sensor nodes are restricted to a transmission range of approximately 20-30 metres indoors and 75-100 metres outdoor [110], which the sensor nodes might require multi hops to reach the destination sensor node. The routing protocols are required to manage and maintain the routes to ensure reliable communications between the limited range nodes. The routing protocols in WSNs are different than the traditional routing protocols as sensor nodes have limited processing capabilities, limited storage and use different operating systems. In order to prolong the network lifetime, the routing protocol plays a role in balancing the energy consumption distribution in the network. Routing protocol should spread the communications load evenly. This helps to avoid certain nodes from being overused, such as the node closer to the root node [28].

2.5.1 Introduction

In WSNs, flooding and gossiping are two classical approaches to relay data to the destination [2] as shown in Figure 2.12. Gossiping is a slightly enhanced version of flooding. These data delivery approaches do not depend on any routing protocols and network topology. In flooding, the packet is broadcast to all the node's neighbours and the neighbours that receive the packet will continue to broadcast the packet to their neighbours until the packet destination is found or the maximum number of hops allowed is reached as shown in Figure 2.12(a). In gossiping, instead of flooding to all the neighbours, it selects a random neighbour to forward the packet. The node that was selected will pick another random neighbour to continue sending the packet until it arrives at the intended receiver. Flooding and gossiping approaches are easy to implement. However, the approaches cause duplicated messages due to overlapping nodes that receive and send the same packet. The approaches also consume a large amount of energy. Gossiping causes propagation delay as the nodes selected are not guaranteed to be the nearest node to get to the destination node. An example is shown in Figure 2.12(b), where the sender node, S sends to node A, B, C and the destination D when it can send to node C then D, which is a shorter route.

The routing protocol is important in ensuring reliable packet delivery in WSNs. Several crucial criteria in the design of a routing protocol are in terms of scalability, reliability, power consumption and adaptability. In WSNs, sensor nodes are typically densely deployed in the error prone wireless channels. This places the importance of scalability in the routing protocol to reduce conflict between the nodes and external interference from other devices while maintaining reachability between nodes. The routing protocol should consider the sensor nodes' battery levels before deciding on the routes. It can prolong the network lifetime by considering alternative routes in order to avoid draining lower energy nodes. Throughout the network lifetime, the nodes may fail, join or move to a different location than the nodes were initially. The routing protocol should be able to adapt to these changes and update the routes accordingly.

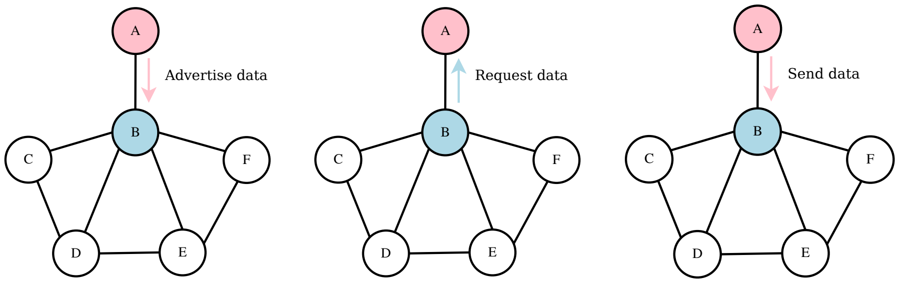
Routing protocols are aimed to provide optimised, scalable and energy-efficient routes in the network, which as a result, could prolong the network lifetime.

2.5.2 Classification of Routing Protocols

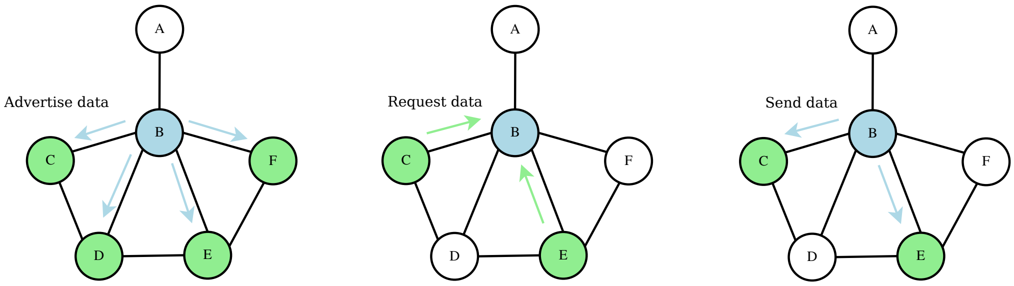
There have been many routing protocols that were developed for WSNs. The routing protocols can be classified into 4 types [2]; (i) data centric, (ii) location based, (iii) network flow and Quality of Service (QoS) aware, and (iv) hierarchical based. These classifications are explained below with examples of the existing routing protocols for each type.

2.5.2.1 Data Centric

Data centric routing is a neighbour-to-neighbour query based routing. It uses attribute-based name or meta data to refer to a specific data item. The sender sends data queries to a certain region and waits for the sensor nodes to send a reply with the queried data. Sensor Protocols for Information via Negotiation (SPIN) [45] and Directed Diffusion [53] are a few examples of the earlier data centric protocols.



(a) Node A advertises data (b) Node B requests the advertised data (c) Node A sends the data



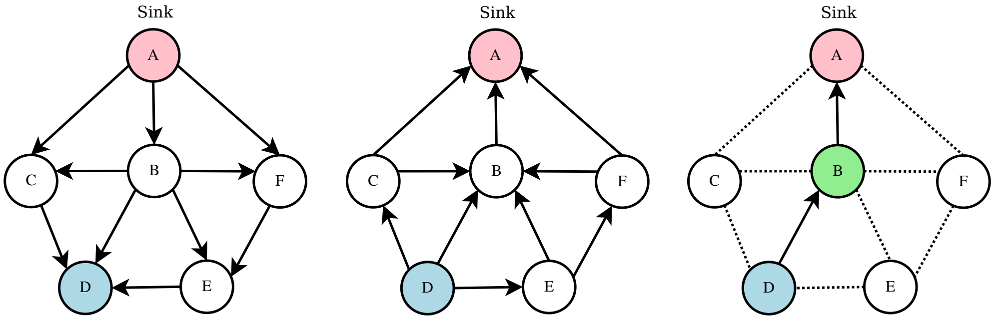
(d) Node B advertises the data from node A (e) Interested nodes C and E request the data (f) Node B sends the data to node C and E

Figure 2.13: SPIN protocol

SPIN is a data centric protocol. It uses high-level descriptors or meta data to refer to the data. These meta data are exchanged using the data advertisement mechanism among the sensor nodes before transmission. Each node in SPIN is only required to know its immediate single hop neighbour. This allows topological changes to take place locally. The nodes that receive the meta data will then advertise the data availability to its neighbours. This allows interested nodes to query the data. However, SPIN does not guarantee the delivery of data to the interested node. The data delivery is decided by the nodes that are situated between the source and destination nodes. If the in between nodes are not interested in the data, the data will not be delivered to the destination.

Figure 2.13 shows an example of SPIN, where node A has a data and advertises it to the neighbour, node B. If node B is interested, it sends a request, which node A will then send the data. Node B advertises the data from node A to all of its neighbours. The neighbours that are interested in the data send a request (node C and D) and node B sends the requested data. Node C and F do not receive the data as they are not interested and did not send a request.

Directed Diffusion is a query-driven data delivery model. Directed Diffusion aims at



(a) Sink broadcasts interest (b) Interest data in node D found (c) Establish the selected path

Figure 2.14: Directed Diffusion routing

diffusing data using attribute-value schemes. The data on the sensor is queried on demand using the attribute-value pairs. An interest or task is defined using a list of attribute-value pairs to create a query. Figure 2.14 shows an example of Directed Diffusion. The sink (node A) broadcasts the interest through its neighbours. The interest is cached at the receiving nodes for later use. Caching helps to increase the routing energy efficiency and minimise delay. It is used to compare the received data with the values in the interests. The reply link to the neighbour from which the interest was received is called a gradient. The paths are established between the sink and sources based on the interest and gradient utilisation, where one of the paths is selected as reinforcement. The interest is then resent by the sink using the selected path with a smaller interval, which in the example is through node B. Directed Diffusion selects a new or alternative path that sends data in lower rates when the current path fails.

In Directed Diffusion, the sink queries the sensor nodes for the specific data. SPIN however, advertises the available data to allow interested nodes to query the data. There are many other protocols that have been proposed based on Directed Diffusion, such as Rumor Routing [16], Gradient-Based Routing (GBR) [95], Constrained Anisotropic Diffusion Routing (CADR) [24]) and other protocols that followed a similar concept, such as Threshold Sensitive Energy Efficient Sensor Network (TEEN) [79], which is also a hierarchical-based, and Active Query Forwarding In Sensor Networks (ACQUIRE) [92].

2.5.2.2 Location Based

In location based routing, the sensor nodes' locations are used to estimate the energy consumption between the distances of the two known nodes. As the location of the nodes is known, the number of transmissions required can be reduced, as the transmissions can now

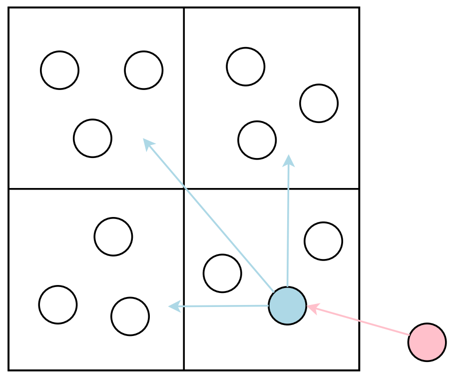


Figure 2.15: GEAR protocol

be targeted to the specific region. However, the nodes are required to be equipped with Global Positioning System (GPS) to allow location of the nodes to be detected. Geographic and Energy-Aware Routing (GEAR) [129] and Geographic Adaptive Fidelity (GAF) [126] are two examples of location based routing.

GEAR is an energy efficient routing protocol that is used to query the targeted regions. The sensor nodes are equipped with GPS to enable location detection. The idea is to restrict the number of transmissions by only considering a certain region rather than the whole network. The nodes are location and energy aware. The nodes are also aware of the neighbour's residual energy. Each node keeps an estimated cost, which is the residual energy and distance to the destination. This enables GEAR to use this information to select the nodes to route a packet to the destination region efficiently. A node sends a packet to the target region. When the node receives a packet, it checks if there is a neighbour that is closer to the target region than itself. The nearest neighbour to the target region is selected. If the packet has reached the targeted region, the packet can be diffused by recursive geographic forwarding or restricted flooding to reach all nodes in the region. This is shown in Figure 2.15. In recursive geographic forwarding, the region is divided into four sub regions. The packets are made into four copies and forwarded into the regions.

GAF is another energy-aware location based routing. It turns off unnecessary nodes in the network without affecting the routing coverage to conserve energy. Figure 2.16 shows an example of GAF. GAF forms a virtual grid for the covered area, where nodes in the same grid (in the example, node 2, 3, 4) are considered to have equivalent cost for routing. The nodes are grouped into virtual grid according to the nodes' location indicated by the GPS.

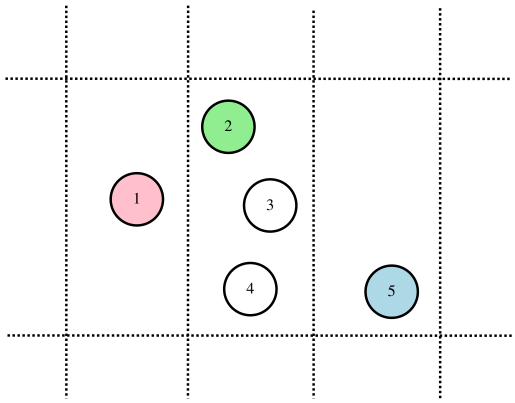


Figure 2.16: GAF protocol

Some of the nodes in the same virtual grid turn off their radio to save energy and wake up before the currently active node expires and goes to sleep. As an example, node 2 is the active node while node 3 and 4 go to sleep. Node 3 or 4 will become the active node before node 2 becomes inactive. GAF keeps a representative node awake on each virtual grid for routing. GAF increases the network lifetime as it exploits the location of the nodes in order to minimise the number of awake nodes in each grid to conserve energy.

2.5.2.3 Network Flow and QoS-aware

In network flow and QoS aware routing, the route setup takes into account the network flow problems, such as network congestion, and the quality of service requirements, such as the end-to-end delay, routes reliability and fault tolerance in routing. Sequential Assignment Routing (SAR) [105] and maximum lifetime energy routing [20] are examples in network flow and QoS aware routing. These routing protocols try to find a balance between energy consumption and QoS requirements.

The maximum lifetime energy routing solution has the objective of maximising the network lifetime by considering the nodes' remaining energy to define the link cost for transmission using the link. The protocol uses Bellman-Ford shortest path algorithm to compute the least cost paths to the destination.

SAR is a protocol that considers the QoS in its routing decisions. The path is selected based on the energy resources, QoS on each path and the packet priority level. SAR proved to consume less energy when it considers the packet priority than other minimum-energy routing that only focuses on the energy consumption. SAR tries to minimise the average weighted QoS metric throughout the network lifetime. SAR maintains multiple paths from

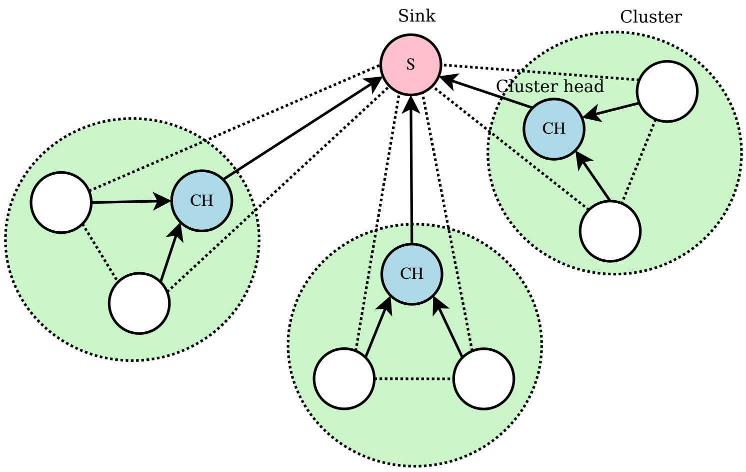


Figure 2.17: LEACH protocol

the nodes to the sink to ensure fault tolerance and easy recovery. However, this resulted in high overhead to maintain the paths in a large network.

2.5.2.4 Hierarchical

Hierarchical based routing aims to scale a large set of sensor nodes that cover a wider area of interest by enabling multi hops communication while maintaining efficient energy consumption. A single hop based routing is not scalable and causes the network to overload when the number of nodes increases, which resulted in conflict in transmissions, thus congestion. Hierarchical based routing usually group nodes into clusters and performs data aggregation and fusion to eliminate duplicate and reduce the number of transmitted messages. Low-Energy Adaptive Clustering Hierarchy (LEACH) [46] is one of the early hierarchical routing in WSNs. The idea proposed in LEACH has inspired many hierarchical routing protocols, such as TEEN [79], Adaptive Periodic Threshold Sensitive Energy Efficient Sensor Network (APTEEN) [78], Power-Efficient Gathering in Sensor Information System (PEGASIS) [72], Hierarchical-PEGASIS [73] and Hybrid, Energy-Efficient, Distributed Protocol (HEED) [128].

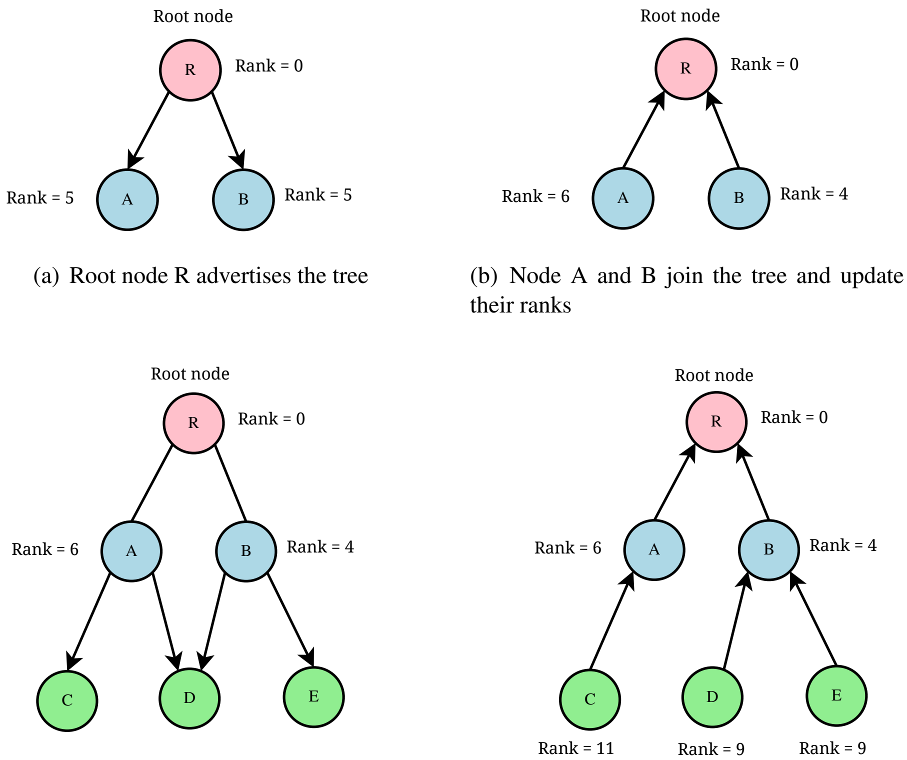
Low-energy Adaptive Clustering Hierarchy (LEACH) is one of the most popular hierarchical routing protocols in WSNs. LEACH forms dynamic clustering of nodes based on the received signal strength and selects a node as the local cluster head. This is shown in Figure 2.17. The cluster head (CH) is used for data processing such as data aggregation and fusion of the nodes within the cluster and to route the processed data to the sink. It consumes a larger amount of energy for data processing. The cluster head changes periodically where the node is selected by random to become the cluster head. This is done in order to balance the energy dissipation of nodes, which increases the network lifetime. The random selection ensures that the nodes die randomly to avoid the network from not functioning. However, it brings extra overhead when the cluster head changes as it has to advertise to the nearby nodes of the changes. LEACH is a distributed routing and does not require global knowledge of the network. It uses a single hop routing where the node transmits directly to the cluster head, and from the cluster head to the sink. LEACH is not applicable to large network that requires multi hops.

Contiki provides support for two hierarchical based routing protocols, Contiki Collect protocol, which is the Contiki implementation of Contiki Collect Protocol (CTP) [44, 41] and RPL [121]. Chrysso uses Contiki Collect protocol as the routing protocol and RPL is compatible with MiCMAC and it is the main routing protocol in MCRP.

Contiki Collect protocol and CTP are data collection protocols that are address free. The nodes send the data usually towards the sink without specifying the node's address. The routing protocol builds a tree originating from the sink and the nodes send periodic announcements, which contain the number of hops away from the sink. Both protocols use the Expected number of transmissions (ETX) as the metric to find the path that requires the minimum number of transmission to reach to the root. The nodes start sending messages towards the root once the tree is built. The messages are sent using hop-by-hop reliable unicast. Contiki Collect protocol and CTP are not IPv6-based. Contiki Collect protocol uses Contiki Rime stack, which is Contiki's lightweight communication stack explained in Section 4.2.1.

RPL is designed largely based on CTP. RPL is a distance vector routing protocol that uses IPv6. It uses a Destination Oriented Directed Acyclic Graph (DODAG), which routes over a tree with a single root. RPL supports any-to-any routing, where the traffic is routed upwards until a common ancestor of the destination and source is found, then downwards to the destination. RPL uses a simple rooted topology instead of a full mesh. It maintains reliable paths to a single destination, which allows RPL to scale to large networks while keeping the routing overhead to a minimum at the cost of increased hop count.

RPL is constructed using an Objective Function (OF) that specifies the routing metric, routing constrains and other functions to construct the topology. The OF is application dependant as RPL does not define any specific OF. There are two OF that are provided by



(a) Root node R advertises the tree

(b) Node A and B join the tree and update their ranks

(c) Node A and B send their DIO to node C, D, E

(d) Node C, D, E join the tree and select lower rank parent

Figure 2.18: RPL building tree

RPL, which are a simple hop count [112], where it selects the path that has the shorter path; and the second OF is the expected number of transmissions (ETX) [43]. ETX depends on the path that requires less transmission to the destination sensor node[117, 114, 115].

RPL has four types of ICMPv6 control messages that are used for topology maintenance and information exchange, which are DODAG Information Object (DIO), Destination Advertisement Object (DAO), DODAG Information Solicitation (DIS) and an optional DAO-ACK message [121]. DIO is the main routing control information that includes the node's current rank, configuration parameters and the root's IPv6 address. DIO allows a node to discover and learn the RPL configuration. DIS is used to enable a node to enquire DIO message from a reachable neighbour. DAO is used to propagate destination information upwards along the DODAG. It also enables down traffic to be supported. DAO-ACK is used as a DAO message response to acknowledge the DAO message by the DAO recipient.

Figure 2.18 shows the RPL tree construction. The topology is constructed from the root node. The distance from the node relative to the other nodes with respect to the root is

Protocol	Scalable	Route Metric	Periodic Message Type	Robust
SPIN	Yes	Single hop neighbour	Advertise to all neighbours	Yes
Directed Diffusion	Yes	Best path	Query messages	Yes
GEAR	Limited	Best route	Hello messages	Yes
GAF	Limited	Shortest path	Discovery messages	Yes
SAR	Limited	Path with minimum average weighted QoS metric	Hello messages	Yes
LEACH	Yes	Shortest path	None	Yes
CTP	Yes	ETX	Beacons	Yes
RPL	Yes	ETX	DIO messages	Yes

Table 2.2: Comparison of studied routing protocols

called *rank*. The rank increases away from the root and decreases when it is nearer to the root. Rank is used to avoid routing loop in the topology as the node's position relative to the other nodes is known. In Figure 2.18(a), the root node sends DIO message to the reachable nodes, A and B. The nodes that receive the message run an algorithm specified by the OF to select a parent. The nodes compute their rank as shown in Figure 2.18(b) and send an update in the DIO message to the neighbours including the root node. Figure 2.18(c) shows new nodes C, D, E joining the network. Both node A and B advertise their DIO message. Node D receives DIO from node A and B. It has to choose a node that has a lower rank as the preferred parent. Figure 2.18(d) shows that node D selects node B and keeps node A as the backup. Node C and E only have one parent option. RPL has rank hysteresis mechanism to avoid frequent parent switching in the case of small rank improvements.

RPL uses the *Trickle* algorithm [65] to control the message sending rate. In RPL, Trickle reduces the control messages rate by exponential increase to avoid the control messages from congesting the network. DIO is sent periodically where the duration is doubled each time a DIO is sent until it reaches Trickle maximum possible value.

2.5.3 Comparison and Discussion

The routing protocols reviewed are summarised in Table 2.2. Based on these, the important factors [86] that influence the routing protocols are:

1. **Node deployment** - Sensor nodes are scattered randomly and require a scalable protocol to allow routes to be formed and adapted. In a large region, the nodes require

multi hops to reach the sink node. While all the routing protocols studies are scalable and robust, location based protocols shown to have limited scalability as the nodes are group into regions based on the location, allowing some of the nodes within the same region to become inactive to conserve energy. The other types of routing protocols use all the nodes in the network for communication.

2. **Route metric** - The routing protocols mostly use shortest path or ETX. This metric selected depends on the application as most protocols allow the user to implement other types of metric. RPL implemented both shortest path and ETX, which ETX is set as the default.

3. **Data delivery models** - The data can be continuous, event-driven, query-driven or hybrid depending on when the data is triggered and the query is generated. SPIN and Directed Diffusion protocols use push and pull method to request and retrieve the data initiated by the root node while the other routing protocols are mostly sender based (sending to the destination node) when there are packets to be sent.

4. **Energy** - The distance from the node to the sink influences the energy consumption in the path. In many routing protocols, the shortest path is used as an indicator to represent energy efficiency as it takes the least number of hops from the node to the sink.

Based on these reviews, the routing protocol is important to ensure high successful communication rate. In WSN, the routing protocol has to be lightweight due to the sensor nodes' limitation and the protocol should be able to perform efficiently under the interference environment. A multichannel protocol enables better paths selection as different channels on the paths would affect the transmissions differently. It would also allow simultaneous transmissions as the nearby routes of different channels do not overlap. To further improve the network lifetime in addition to the multichannel protocol, it is important that the routing protocol is energy aware to ensure that certain nodes and paths are not being overused. MCRP is built based on these observations to improve the existing routing protocol.

2.6 Summary

This chapter describes the examples of existing WSNs applications currently used and deployed to track and monitor in different types of environments. It is increasingly important to have a reliable and energy efficient network that could function for years as nodes are easily deployed in areas that are difficult to reach, such as for volcanic monitoring, forest fire detection and flood detection. In terms of smart cities, WSNs help to enable automated services, which improve the environment quality, increase the energy saving and improve the lifestyle as WSNs simplify and reduce manual and labour work to automated systems. However, WSNs operate in an unreliable radio environment. This increases the chances of packets being lost due to interference. This leads to frequent retransmissions, which could congest the network and drains the network lifetime quicker. In order to overcome this problem, many energy efficient protocols have been proposed in terms of the MAC protocols, routing protocols, power control and energy harvesting to prolong the network lifetime while maximising the throughputs. There have been many proposals in multichannel MAC protocols that show promising results but none is widely implemented yet. In MAC protocol, the energy consumption is reduced by adjusting the radio duty cycle to allow the sensor node to be in the sleep mode when it is not used and to transmit on channels that have less interference to increase the likeliness of data success. In order to further improve the efficiency and ensure reliable communication, routing protocols play the role in optimising and balancing the transmission load to the sensor nodes evenly. This helps to avoid overusing certain sensor nodes as routes, which would drain the sensor nodes' lifetime more quickly, thus creating a non-functional network.

By using multichannel MAC protocol together with a reliable routing protocol, the network lifetime can be prolonged while maintaining high packet reception rate in WSNs. The Multichannel Cross-Layer Routing Protocol (MCRP) is introduced to improve the network performance based on the existing protocols studied. MCRP is designed to take into account the advantages of ContikiMAC's duty cycle and the RPL standard protocol that are lightweight and efficient. MCRP is a multichannel protocol that runs in real-time to allow the protocol to adapt to the surrounding interference at any location. MCRP is explained in detail in the next chapter.

Chapter 3

Multichannel Cross-Layer Routing Protocol

3.1 Introduction

WSNs often suffer from frequent occurrences of external interference, such as Wi-Fi and Bluetooth. Multichannel communications in wireless networks can alleviate the effects of interference to enable WSNs to operate reliably in the presence of such interference. As a result, multichannel solutions can improve the network efficiency of spectrum usage, network stability, link reliability, minimise latency and minimise the number of packet losses, hence, retransmission.

This chapter presents Multichannel Cross-Layer Routing Protocol (MCRP) [82], a decentralised cross-layer protocol with a centralised controller. The cross-layer multichannel protocol focuses on the network and application layers. The system has two parts: a central algorithm which is typically run by the LPBR and selects which channel each node should listen on, and a protocol that allows the network to communicate the decision to change channel, probe the new channel and either communicate the success of the change or fall back to the previous channel. MCRP concentrates on finding channels for the nodes that are free from or have low interference. It allows the allocation of these channels in a way that is likely to minimise the chance of nodes that are physically close communicating on the same channel. Hence, it reduces cross interference between different pairs of nodes.

3.2 MCRP Design

WSNs have the following properties: limited memory and battery capabilities, and it is impossible to determine a single channel that is free from interference at any location. The design of the multichannel protocol is based on these crucial observations:

i. **Channel assignment** - In order to maximise the sensor's lifetime, a centralised LPBR that has a larger memory and is fully powered is used for decision-making. The LPBR has complete knowledge of the topology, which enables it to make good channel assignment decisions based on a two-hops colouring algorithm.

ii. **Interference** - External interference cannot be predicted, therefore channels cannot be allocated beforehand, as it varies over time and location. The protocol checks the channel's condition each time before deciding on a change of channel to reduce interference and maximise throughput.

iii. **Frequency diversity** - Multichannel protocol increases the robustness of the network against interference. However, applying multichannel protocol to the existing RPL may hinder detection of the new nodes and cause problems for maintaining the RPL topology. Two mechanisms are introduced to overcome this problem; these solutions are explained in detail in Chapter 4. Existing nodes maintain a table of the channels on which their neighbours listen on and use unicast to contact those nodes. New nodes listen on the Contiki default channel (26) and, when connecting, the nodes search through all channels to find the neighbouring nodes. As in RPL, periodically all nodes broadcast the RPL control messages on the default channel in an attempt to contact new nodes.

MCRP is built, based on these observations, to overcome the shortcomings of sensors and sensor networks by concentrating on multichannel protocol and strategies to avoid channels with interference.

3.3 MCRP Overview

MCRP is a cross-layer multichannel protocol, concentrating on three layers: the application layer, network layer and MAC layer. Figure 3.1 shows the different layers, examples of the layers and the cross-layer functionality, where different layers can access the information from the other layers as necessary. The application layer refers to the high-level layer where the user can program the node to run processes. The application layer is normally used to initiate normal data transmissions to the other nodes. The sensor node applications can be dependent or independent of the sensors in the node. The node used in this thesis is TelosB [110]. TelosB has three types of sensors attached, light, humidity and temperature sensors, and the sensed values can be accessed on the application layer for specific experiments, such as to plot graphs and for monitoring purposes.

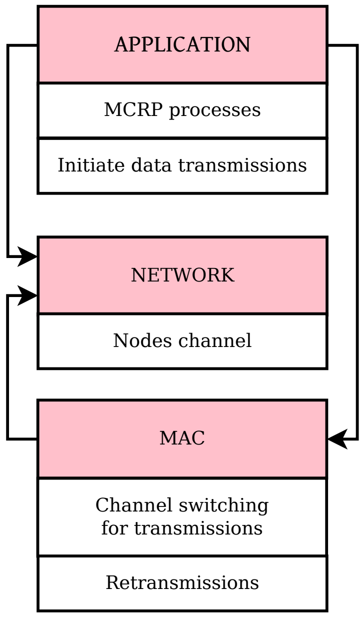


Figure 3.1: MCRP cross-layer protocol

In MCRP, the sensors are not used. The protocol implements MCRP processes on the application layer, such as the channel selection through two-hops colouring algorithm, channel quality checking and the channel decision-making processes. The information on the channels for the neighbouring nodes is stored in the network layer, which can be retrieved by the application layer (to update the channel value in the network layer if there are any changes) and also from the MAC layer. The MAC layer requires access to the network layer in order to enable correct channel switching for data transmission and retransmissions for the specific node.

MCRP consists of a centralised controller (referred to as the LPBR) and a decentralised part (also referred to as the other nodes) shown in Figure 3.2. There are three main parts of MCRP: the centralised channel selection strategy, the decentralised channel switching and the decentralised channel quality checking. The centralised controller is responsible for the channel selection algorithm (based on the two-hops colouring algorithm) and for storing MCRP results, while the decentralised nodes take part in the channel decision-making steps through channel quality checking on the current interference environment and to implement the channel decisions. The intelligence of MCRP in channel selection for all the nodes in the network is done at the centralised LPBR, as the individual nodes have limited capabilities due to the nodes' constraints in batteries and memory availability.

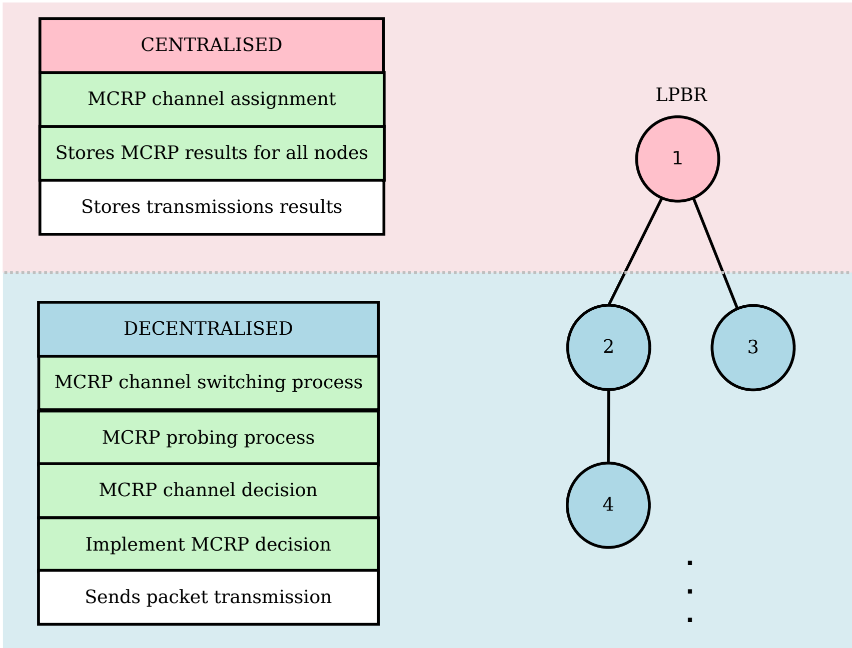


Figure 3.2: MCRP centralised and decentralised nodes

MCRP is a dynamic on-line protocol where the channel's condition is checked in real-time when the protocol is invoked, before the decision of switching to a new channel is finalised. Existing multichannel protocols usually have a fixed list of channels to switch into at each iteration (referred to as off-line), that the channels might be checked before it is included in the list. The channels could be selected at random. This depends on the protocol itself. MCRP dynamic on-line protocol has the advantage of being able to adapt to any location, as the protocol checks the channel's condition before making the decision for the node's channel switching.

The rest of this chapter describes in detail MCRP (i) channel selection strategy, (ii) channel switching decisions, (iii) channel quality checking and (iv) reconnection strategy.

3.4 Channel Selection Strategy

One main advantage of MCRP is generality. Any algorithm can be used at the LPBR to assign channels. MCRP uses a two-hops colouring algorithm to select a channel to be assigned to a node. The two-hops colouring algorithm attempts to ensure that nearby nodes do not communicate on the same channel and risk interfering with each other. The protocol is inspired by the graph colouring problems [57]. The core idea is that no node should use the same listening channel as a neighbour or a neighbour of a neighbour (two hops). This allows fair load balancing on the channels and reduces channel interference that could

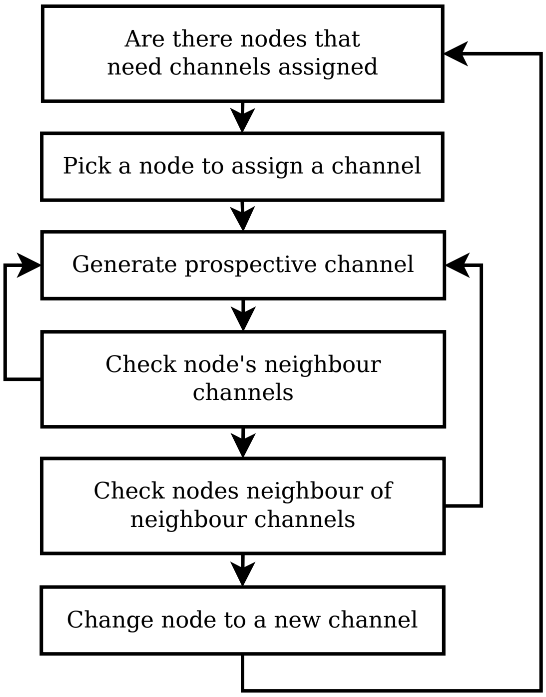


Figure 3.3: Channel selection strategy

occur when two nearby nodes transmit together on the same channel. The nodes used in this experiment have a transmission range of approximately 20-30 metres indoors and 75-100 metres outdoors [110]. It could be the case that many nodes in a sensor network are in the transmission range of each other and potentially interfered with.

All nodes are initialised to channel 26, which is the common default channel for Contiki MAC layer since it often has fewer interference problems with Wi-Fi and other sources. The studies in [56, 5, 119] use a set list of whitelisted channels in their experiments and have channel 26 in common. However, the initial channel should not be used as the transmission channel throughout the runs (unless all channels check fail) in order to not overload the initial channel, thus, the change to other interference-free or low interference channel. It could also lead to interference between the nodes competing for transmissions opportunity, which is the problem in a single channel network. To ensure reliable communication, two-hops colouring algorithm is used in channel selection to avoid the selected channel from overlapping with the other nearby transmissions.

The default RPL setup mechanism is used to exchange control messages that are required to form an optimised topology before channel assignments can take place. The nodes will only be on the same channel once during the initial setup. This enables the node to detect and find nearby neighbours that are in range before it decides on the best route based on the list of neighbours it can be connected to.

Algorithm 1 Pseudo-code for two-hops colouring algorithm

```
Notations
R is a node that is a routing node
RN is the Route's Neighbour node
RNN is the Neighbour node of the Route Neighbour RN
currentCh is the node's current listening channel
newCh is the new channel the node will change to
Pseudo-code
Check if there are nodes which need a new channel
Generate a newCh by random for the node R
if R currentCh ≠ newCh then
    check all RN channels
    if RN channel ≠ newCh then
        proceed one-hop
        check all neighbours of RN
        if RNN channel ≠ newCh then
            proceed two-hop
            confirm newCh
        end if
    end if
else
    generate a new newCh
    update the number of newCh generated for R
    if number of newCh generated > 3 then
        use default channel 26 if all tries fail
    end if
end if
```

The pseudo-code of the implemented two-hops colouring algorithm for the new channel selection is shown in Algorithm 1. When the new channel is selected, the LPBR will send the channel value to the intended node. To clarify, a routing node refers to the node that is used to transmit the packet to the next hop node (or the intended destination). Nodes that are in range of each other are referred to as neighbour nodes. Not all neighbour nodes are routing nodes. The neighbour nodes are selected as routing nodes based on the RPL ETX values (nodes on the routes that take the least expected transmissions rate, meaning less interference on the route).

In the two-hops colouring algorithm, the LPBR chooses a node to which it will assign a channel to listen on. The selection is random (from channels 11 to 26) based on the full range available [48]. The authors in [107] proposed a spectrum sensing algorithm to decide on the number of channels to be sensed before the channel is selected for transmissions. While it was found that sensing more channels increase the likeliness to find the best channel with less or no interference, it requires higher energy consumption and longer time delay. In the studies, three cases were considered where (i) all channels are sensed, (ii) the energy consumed (channel's condition) in all channels is known and (iii) in the case where an energy threshold is set in unknown conditions.

Instead of checking all or several channels, MCRP chooses a random channel and learns the channel's condition from the current simulation. The channel's condition might vary depending on the location and time, and it could be the case where the same channel has different interference level each time. In this case, knowing the channel's condition does not have any benefit and requires all or several channels to be rechecked each time. This consumes a lot of energy during each check. MCRP only considers two channels at a time, whether the new channel has better reception rate than the current channel. By doing so, the channel selection is more spread out (random and fulfil the two-hops colouring rule) rather than all nodes trying to use the same best channel.

The channels that were tested to have severe interference for one node might give good results for another node depending on the location of the node, which might not be within the range of where the channel has severe interference previously. MCRP has its channel quality checking mechanism before it decides on a channel.

The protocol checks the node's neighbours and neighbours of neighbours to see if any of those are listening on this channel already. If any are, a new channel is picked from the remaining list of available channels. If the LPBR has knowledge of existing bad channels, then those channels can be blacklisted. Knowledge of the channel interference, which is gained by probing can be used to decide that a channel should not be used. If a channel is found, then the channel switching protocol is triggered. If no channel can be found meeting these conditions, the current channel is kept. Figure 3.3 summarises the strategy in LPBR channel selection.

The node's selection algorithm must only attempt one channel change at a time to ensure probing is done on the correct new channel and for the node to finalise the channel to be used before another node attempts a channel change. The protocol ascertains that the channel change attempt will always result in a message returned to the LPBR either

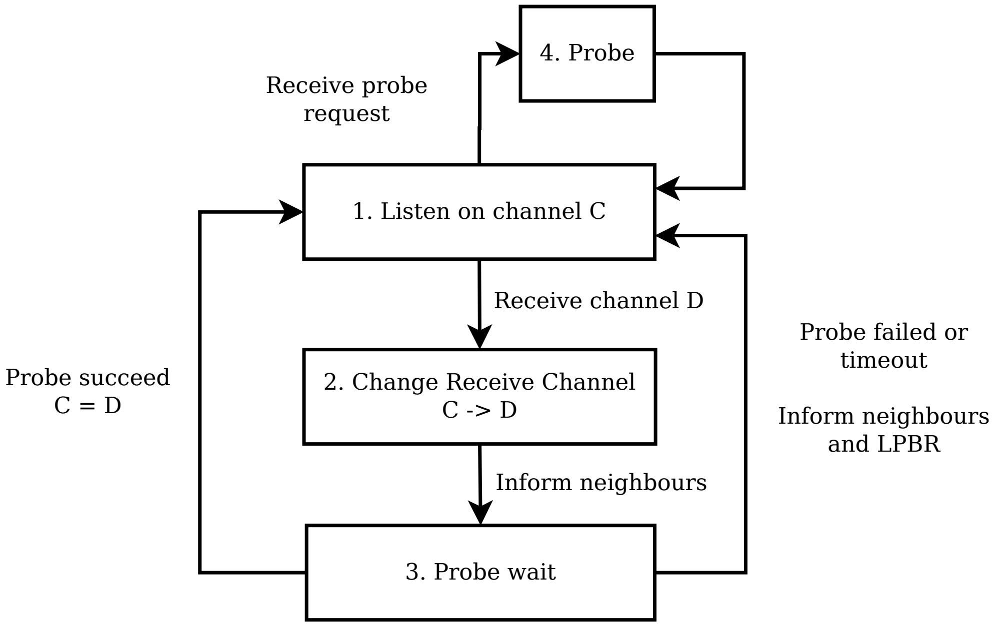


Figure 3.4: Channel switching processes

confirming the new channel or announcing a reversion to the old channel. Until one or other of these happens, no new channel change will be made to ensure that the neighbours are transmitting on the correct channel.

3.5 Channel Switching

Figure 3.4 shows the state machine for the channel switching protocol. As explained in the previous section, a choice of a new channel by the channel selection protocol causes the change channel message to be sent to the appropriate node. Upon receiving the channel change message, the node N stores its current channel C and communicates to all its neighbours the new channel D that it wishes to change to. Those neighbours will update their neighbour tables to ensure that they now send to node N on channel D. The node N begins the channel quality checking process with each neighbour in turn by sending them a probe request. If this process fails for any neighbour then the node reverts to channel C. Node N informs its neighbours of the decision. However, if all channel quality checks succeed, the node N will listen on channel D. Node N does not send a confirmation message to the neighbours as it would be redundant since the neighbours already know node N listening channel. In both cases, the LPBR is informed of the channel checking results. The LPBR updates the node's channel on its table. The channel checking process uses probe packets that might interfere with other transmissions temporarily. However, it is important to emphasise that the network remains fully functional and connected at all stages of this protocol.

3.6 Channel Quality Checking

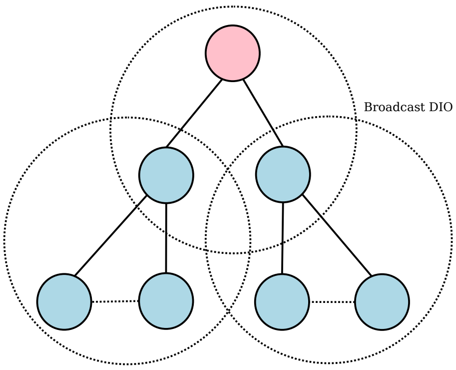
The channel quality checking is invoked each time a node changes channel after receiving a message from the LPBR. The node *N* changing to channel *D* informs all neighbours in turn, of the new channel *D* it will be listening on as described in the previous section. It then enters the *Probe Wait* state and begins channel quality checking with each tree neighbour in turn. In describing the channel quality checking process, it is worth emphasising the distinction between the neighbours and the tree neighbours. The node neighbours are all nodes that a given node knows it could transmit to. The nodes are within the transmission range of each other. The tree neighbours are the nodes that a node does transmit to through the topology formed by the RPL protocol. The tree neighbour node is selected from the list of available neighbours based on the ability to transmit to the next hop towards the LPBR depending on RPL. By default, it is decided by using the least expected number of transmission from the node to the LPBR.

In the *Probe Wait* state, node *N* sends a *Probe* message to each tree neighbour in turn. The neighbours respond to the message by sending eight packets to node *N* on the new channel *D*. The buffer can accommodate eight packets at a time. As the packets might not be sent immediately due to wake up and collisions, sending more packets would have the risk of being dropped. The authors in [97] observed that a short period of time is sufficient to give an overview of the channel's condition as increasing the period shows minimal benefit. The condition of the channel is further investigated through the number of retransmissions and packet collisions of the probing packets for accuracy of the channel's condition.

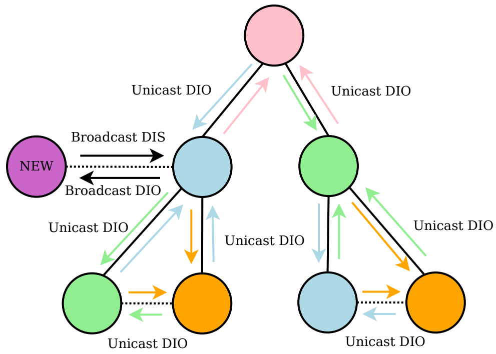
If the probing process times out (because of some communication failure) or the number of probe packets received is above a threshold (currently set to 16, including retransmissions and collisions) then node *N* immediately exits *Probe Wait* state and reverts to channel *C*, its previous channel.

All neighbours are informed of the change back to channel *C* and the LPBR is informed of the quality check failure with a summary of all probes received. If, on the other hand, all channel quality checks succeed, channel *D* is used for node *N* channel. Node *N* informs the LPBR of the results of the probing (numbers of packets received) and the channel change.

Probing is essential to make the channel change decision. It gives a quick overview of the channel's condition based on the number of probing messages received. Probing is



(a) Default RPL control messages as broadcast



(b) MCRP RPL control messages as unicast and broadcast

Figure 3.5: RPL control messages transmission

only done between the node and the tree neighbours. The neighbours that are not the tree neighbours will not use the node as a route during their transmission thus, there is no need for probing to take place with those neighbours. However, the neighbours still need to know the channel value given that the RPL control messages are sent to the neighbours directly without using the routes.

3.7 Reconnection Strategy

RPL is typically used in the single channel protocol. It can be used in multichannel protocol. Figure 3.5 shows the communication method used by the default RPL and MCRP RPL in transmitting the control messages. Figure 3.5(a) shows that RPL broadcasts the control messages to the nearby nodes. According to the RPL standard [121], the RPL control messages can support broadcast and unicast communication. RPL for MCRP is modified to allow communication using unicast and broadcast to support the nodes that might be on different channels. The RPL functionality remains the same despite the modification. Figure 3.5(b) shows the modification where each node sends the RPL control messages on the transmission channel to the specific receiver.

If a new node tries to join the topology, it sends a broadcast of the RPL control message (DIS) through all channels as the listening nodes are unlikely to be on the default channel. The listening nodes send a DIO message as a broadcast on the default channel to discover new nodes (in Contiki default, new nodes will start on channel 26) and send the RPL messages through unicast when the neighbours are known to reduce unnecessary transmissions in broadcast. New nodes and nodes, which have previously left the network can now rejoin on many potential channels.

3.8 Summary

This chapter introduces MCRP, a multichannel cross-layer routing protocol that jointly optimise the network communication through channel decisions from the application, network and MAC layers. It uses a centralised controller for nodes' channel allocation. The channel's condition is tested through probing packets during run time as it varies depending on the location, time and usage. By doing so, the nodes would use less interference channels for communication to maximise the packet reception rate. MCRP uses a two-hops colouring strategy to avoid nearby nodes from competing to transmit on the same channel at the same time, and internal interference that could occur. In order to allow reconnection and new nodes to join an existing topology, the RPL control messages are adjusted to support multichannel protocol.

Chapter 4

MCRP Implementation

4.1 Introduction

MCRP is implemented on the TelosB [110] mote platform. It uses Contiki [29], a lightweight Operating System (OS), as the software development platform that supports the standard IPv6. The implementation of MCRP across the layers in Contiki is described in detail, specifying the changes that were introduced and undertaken in addition to the default parameters and settings in Contiki.

4.2 Contiki

Contiki is defined by a network stack of four layers, as shown in Table 4.1: the network layer, the MAC layer, the Radio Duty Cycle (RDC) layer and the radio layer. The network layer includes support for Transmission Control Protocol (TCP), User Datagram Protocol (UDP), Internet Protocol version 6 (IPv6), Internet Protocol version 4 (IPv4), RPL routing protocol and 6LoWPAN. IPv6 over Low Power Wireless Personal Area Networks (6LoWPAN) is a header compression and fragmentation format for IPv6 packet delivery over IEEE 802.15.4 networks [47]. Contiki implements a minimal set of IPv6 protocol features and

<table>
<tr><th>Contiki</th><th>IoT/IP</th><th>Applications</th></tr>
<tr><td rowspan="4">Network</td><td>Application</td><td>HTTP</td></tr>
<tr><td>Transport</td><td>TCP, UDP</td></tr>
<tr><td>Network, Routing</td><td>IPv6, IPv4, RPL</td></tr>
<tr><td>Adaptation</td><td>6LoWPAN</td></tr>
<tr><td>MAC</td><td>MAC</td><td>CSMA/CA</td></tr>
<tr><td>RDC</td><td>Duty Cycling</td><td>ContikiMAC</td></tr>
<tr><td>Radio</td><td>Radio</td><td>IEEE 802.15.4</td></tr>
</table>

Table 4.1: Contiki network stack

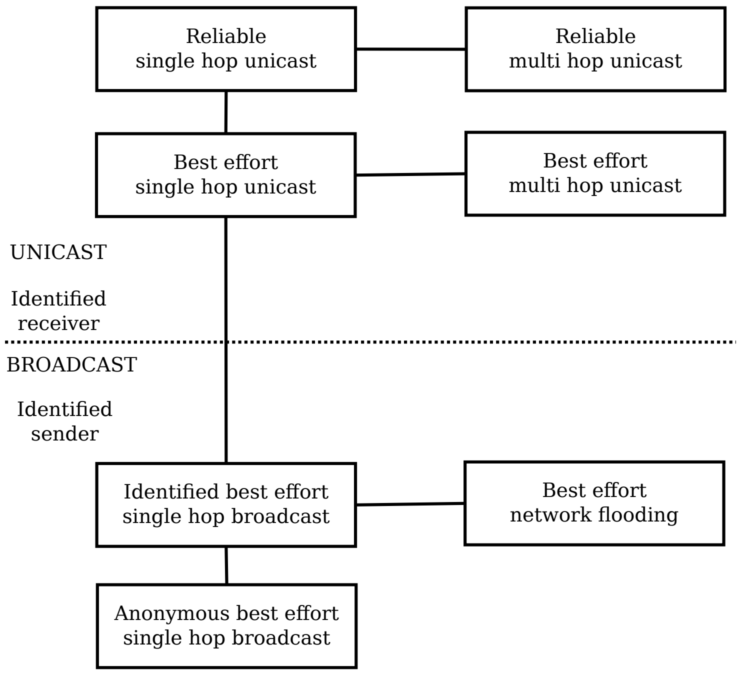


Figure 4.1: Rime stack communication layering

the 6LoWPAN adaptation layer for IPv6 header compression and fragmentation, which is routed over the Low Power and Lossy Network (LLN) in RPL. Contiki's configuration options for communication, buffer management and network interface are explained below.

4.2.1 Communication Stacks

Contiki contains two communication stacks, uIP and Rime. Micro Internet Protocol (uIP) [30] is a small RFC-compliant TCP/IP stack that is designed to contain only the essential features to provide Contiki with TCP/IP networking support, to allow Contiki to communicate over the Internet, compared to the traditional TCP/IP that requires high resources to fit in the limited RAM capability of a sensor. The minimal set of features includes IP, ICMP, UDP and TCP protocols. The uIP deals with the TCP and IP protocols [25, 26].

Rime is Contiki's lightweight communication stack that aims to simplify the sensor network protocol implementation by reusing codes in a layered manner [31]. Rime combines layers of simple communication abstractions to form a powerful high-level abstraction ranging from the best-effort anonymous broadcast to reliable network flooding. Figure 4.1 shows part of the communication layers in Rime. An as example, a reliable multi hop unicast communication consists of the combination of best effort single hop broadcast, best effort single hop unicast and reliable single hop unicast. Parts of the Contiki lightweight

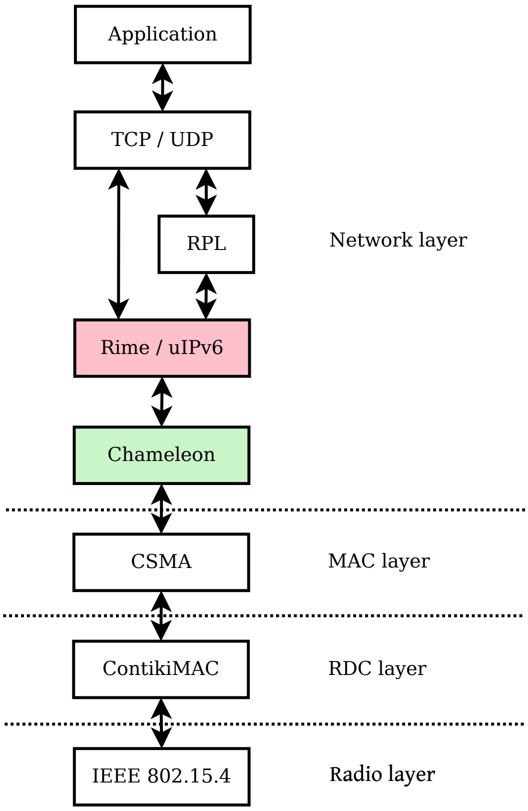


Figure 4.2: Contiki stack

communication stack (Rime) can be used by the underlying MAC or link layer. Additional protocols that are not in Rime can be implemented on top of the stack, such as TCP/IP.

Applications in Contiki can decide to use either one, both or none of the communication stacks available. Rime is an alternative option for simpler communication without the need for IP addresses, which means the nodes can be labelled locally (e.g. numbered) but not globally, as the labelling can overlap. uIP can run over Rime and similarly, Rime can run over uIP [32]. This allows protocols or applications that are not in Rime or uIP to be implemented directly on top of the communication stack, such as running TCP/IP on Rime.

4.2.2 Buffer Management

Chameleon [34] is a communication architecture in Contiki that consists of the Rime communication stack and a set of packet transformation modules. It uses an abstract representation of the information, which allows access to the low-level features of the underlying MAC and link layer protocol from the applications and layers implemented on top of the Chameleon architecture, as shown in Figure 4.2. It also allows the output from the proto-

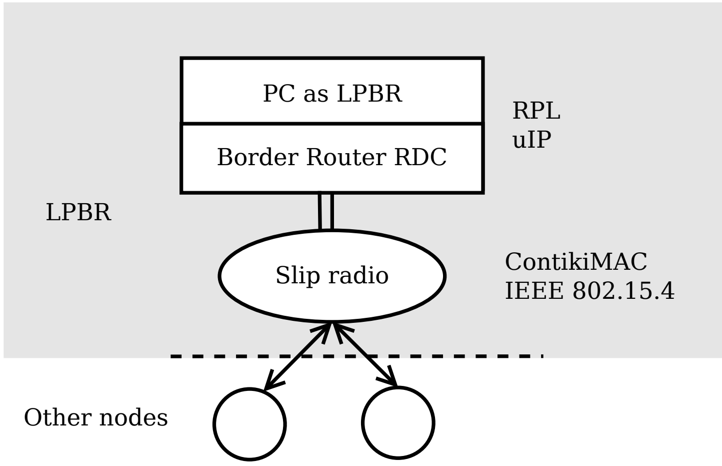


Figure 4.3: MCRP implementation

col stack to be adapted by other communication protocols. In Chameleon architecture, the parsing of its header is separated from the communication stack. This allows uIP or Rime communication stack to be used as described in Section 4.2.1. Chameleon architecture enables the layers to access information without violating the layering principle.

In the buffer management module of Chameleon architecture, all incoming and outgoing packets from the applications and packet attributes are stored in a single buffer called the Rime buffer [30, 34, 31, 29]. All layers of Contiki's network stack including uIP, Rime and the underlying link layer operate on the same packet buffer for the buffer management. The Rime buffer has no locking mechanisms as it is a single priority level buffer. The buffer only holds the current packet.

Protocols that need to queue packets allocate the queue buffer dynamically. The queue buffer is used to hold the queued packets, such as for MAC protocol that has a high rate of incoming and outgoing messages before it can send or process the receiving packets, or when the radio is busy and the MAC protocol has to wait for the radio medium to be free before proceeding with transmissions. The Rime buffer contents are copied into the queue buffer when there is a queue buffer allocated.

4.2.3 Tunslip

Serial Line Internet Protocol (SLIP) [91] is a protocol that has a low complexity and small overhead commonly used to encapsulate IP packets for point-to-point communication between the sink (LPBR) and the device connected, such as an embedded PC across the serial connection. The communication between the devices can take place on any reliable network, such as the Ethernet where the LPBR can be connected to an embedded PC, which contains an Ethernet interface as shown in Figure 4.3 labelled *Slip radio*.

Contiki provides support to communicate with devices using SLIP through its tunslip tool. Tunslip is used to bridge the IP traffic between the LPBR and the embedded PC over a serial line. The other side of the serial line does a similar job to bridge the embedded PC to the LPBR using the network interface. It constructs a SLIP tunnel between a virtual network interface (tun) and SLIP, the physical serial interface to encapsulate and pass the IP traffic to and from the other side of the serial line. The tun interface is used as any real network interface, such as for routing and traffic forwarding [54, 1].

Figure 4.3 shows the overall view of MCRP. The protocol implementation is separated into two types of nodes: (i) the centralised LPBR where the bridging (Tunslip) takes place between the border router on a PC to the nodes, and (ii) the decentralised transmission nodes referred to as the other nodes.

4.3 MCRP Implementation

MCRP is implemented in Contiki and uses ContikiMAC as the MAC protocol and RPL as the routing protocol. ContikiMAC is modified to allow multichannel where the channel selection processes take place on the upper layer and the channels are kept in the network neighbour table to ensure the correct channel for communication.

As explained in Chapter 3, the protocol implementation is separated into two types: (i) the LPBR implementation (centralised) and (ii) the other nodes (decentralised). The implementations for both types are described below.

4.3.1 Low Power Border Router

As sensors have limited memory, most processing decisions at the LPBR are transferred to a PC as it has more RAM and better processing capabilities. This enables MCRP to perform more complex processing and to run in real-time without draining the memory and battery on a sensor. The LPBR is divided into two main parts as shown in Figure 4.4 where the PC is responsible as the application, transport, network and routing layers while a sensor (labelled slip radio) is set as the wireless interface to enable the PC to communicate with the other nodes via the Contiki tunslip tool. The other nodes contain all the layers.

The LPBR acts as the root tree in RPL, where it will initiate the creation of the RPL routing tree. The LPBR is a special case as the channel changes at the LPBR is not as direct

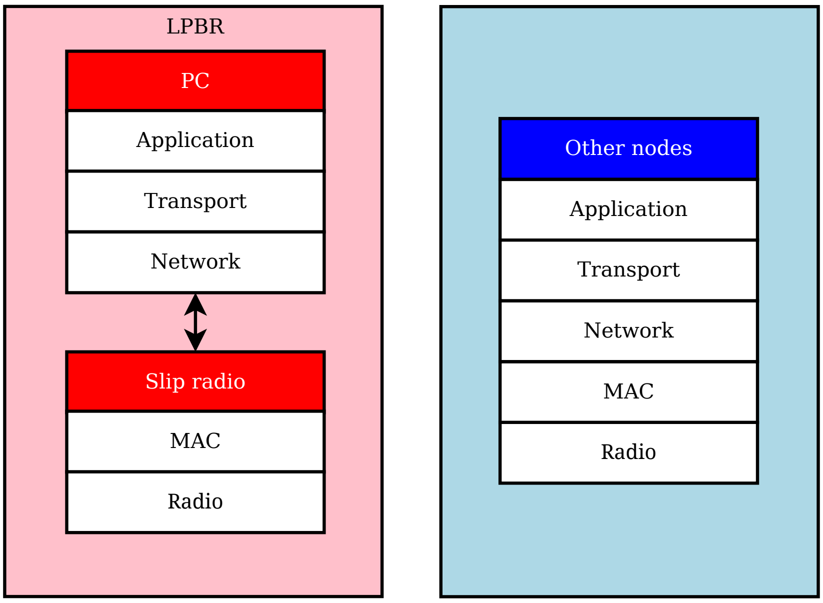
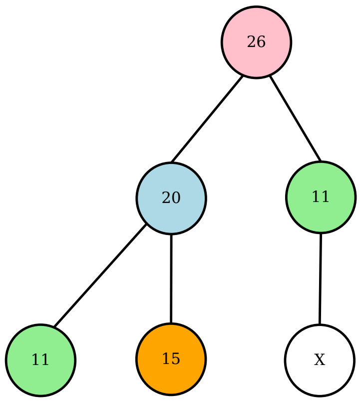


Figure 4.4: Nodes stack

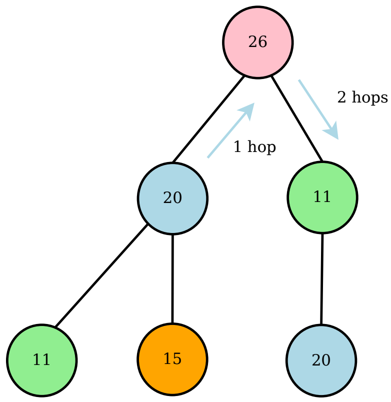
as the other sensor nodes due to these two parts (PC and slip radio). However, it works similar ways to the other nodes. The LPBR's main responsibility is to decide on the new channel selection for the nodes. The LPBR has no knowledge of all the channels' condition at this point, thus, a channel is selected at random. The LPBR keeps the results from the channel changes processes. It uses the information to select a new channel for the next node, which the new channel is at least two-hops away from another node using the same channel. This is done to ensure that the nearby nodes do not communicate on the same channel and risk interfering with each other.

Figure 4.5 shows the two-hops colouring algorithm before a channel is selected for a node in MCRP's channel change process. In the example, the node X could use either channel 20 (blue) or channel 15 (orange) as both channels are at least two-hops away from the other nodes that are using those channels as shows in Figure 4.5(b) and Figure 4.5(c). Channel 26 however, cannot be used as it is not two-hops away (excluding the second hop meaning the channel can be used if the node is at least, on the third hop) from the other node as shown in Figure 4.5(d).

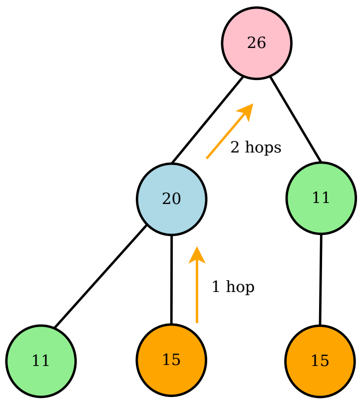
The new channel is stored in the Rime buffer before the data is sent over SLIP to the radio-chip (slip-radio). As the slip radio is unable to access the neighbour table where the next hop node's channel is stored (due to the LPBR being separated into two parts),



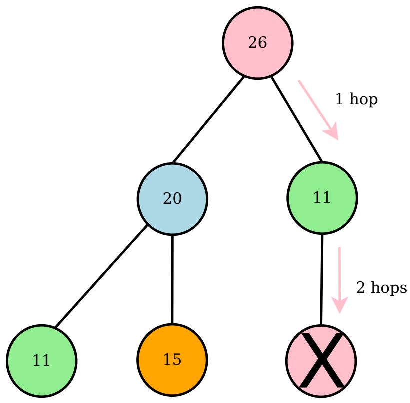
(a) The LPBR checks node X two-hops channel



(b) Channel 20 can be used for node X



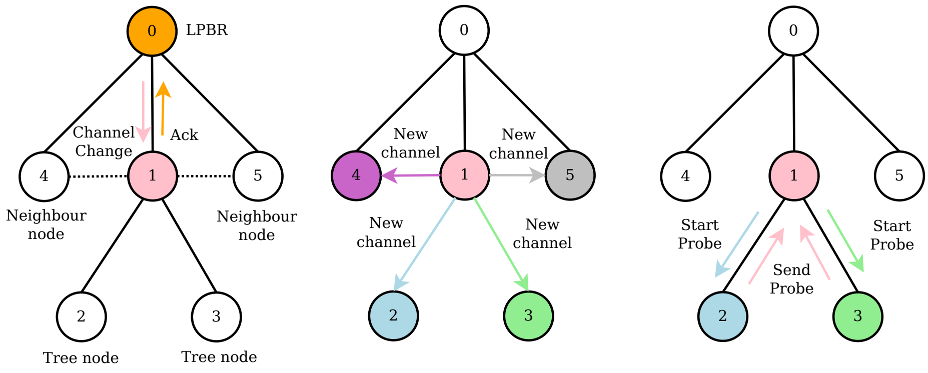
(c) Channel 15 also can be used



(d) Channel 26 cannot be used

Figure 4.5: Two-hops channel checking

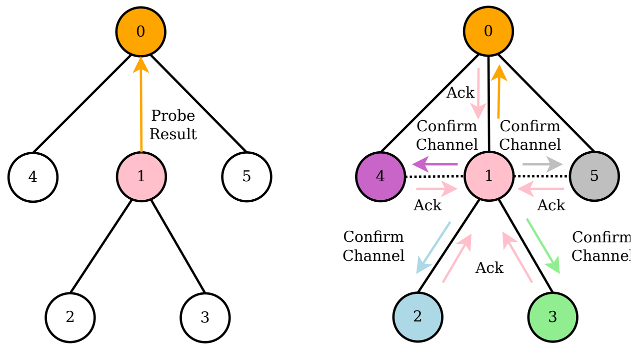
the channel value is attached to the data. The LPBR keeps the updated value of all its neighbours' channels in the neighbour table. Slip radio that receives the data can access the channel value that was attached, and keeps the channel value in a simplified version of the neighbour table in the node's RAM. This is done in order to ensure that the packet that is being queued or retransmitted is sent on the correct channel. The packet destination, which in this case, the next hop node is first checked before the packet is transmitted each time. The MAC layer sets the channel accordingly before sending. ContikiMAC can access the simplified neighbour table as it is on the slip radio. The simplified neighbour table only keeps the information of the node's neighbours and the neighbours' channels, which are the critical information in order to transmit packets correctly. The other information that is related to the neighbours' conditions is monitored at the PC.



(a) The LPBR decides a channel change for node 1

(b) Node 1 informs all the neighbouring nodes

(c) Tree nodes send probes in turn to node 1



(d) Node 1 sends probe results to the LPBR

(e) Node 1 notifies all neighbouring nodes the confirmed channel

Figure 4.6: Nodes' channel change processes

The slip-radio resets to its listening channel after the packet is transmitted. The LPBR will wait and listen to any incoming packets. In the channel probing phase, the LPBR does not take part in probing. However, the LPBR is informed of the results of probing and keeps a table of the probing results and channels to be able to use the information when deciding on a channel change based on the previous results of probing on the known channels.

4.3.2 Other Nodes

Figure 4.6 shows the processes during the channel change. The new channel from the LPBR that is received by the destination node is saved. When the LPBR sends a *Channel Change* message to the destination node, the destination node will send a packet back to the LPBR to acknowledge the channel change message as shown in Figure 4.6(a). If the LPBR does not receive the message, the channel change message is retransmitted. The LPBR will then wait and listen for any incoming packets. At this point, the channel changes processes will take place between the node and its neighbours. To clarify, the *node* refers to the node that would like to change its listening channel and the *neighbour node* is the neighbour of the node. The channel change decision is done through probe messages between the node and the tree node (node that is the routing node). The neighbour node is informed of any channel changes to enable control messages between the nodes (which are sent directly without using the routing tree) to be transmitted and received at any point of MCRP's channel change processes.

Unlike the LPBR, the other nodes have all the layers within the nodes themselves. This makes channel changes less complicated. However, the nodes are being limited by the amount of RAM they have, which resulted in probing values to be stored in the centralised LPBR (which is the PC with unlimited memory). The nodes however, keep the probing results temporarily before the final decision of the channel is made.

The node sends the *Node New Channel* value to all of the node's neighbours on the neighbours' channels as shown in Figure 4.6(b) where the different colours represent different channels. At this point, the new channel is not yet checked for its validity. However, all the neighbours need to know the new channel as the node will change its listening channel to the new channel. Otherwise, the packets cannot be received by the node since the listening channel is different than it was previously. The neighbours that receive the node's channel will update their *neighbour table*, which can be accessed from the application layer. In the neighbour table, a new entry is added to hold the channel value called *nbrCh*. As this is an important step in order to reduce the number of packets lost due to sending on the wrong channel, the neighbours will send an acknowledgement of receiving the new channel. Otherwise, it will be retransmitted.

The node will then send a *Start Probe* message to the neighbours that is a route node to start sending probing messages on the new channel as shown in Figure 4.6(c) in turn. Not all of the neighbours are used as the routes. The neighbours are chosen as a route based on the RPL OF, which for this experiment is the ETX. The node will listen on the new channel and waits for the *Neighbour Probe* message. The route node starts to *Send Probe* messages every 3 seconds to allow retransmission or collision that could happen due to the busy channel. The maximum number of retransmissions is configured to 3 attempts following the default value that Contiki suggested. Collisions happen when the channel check keeps failing and new packets are constantly generated, which could end up in a loop where no packets can be sent.

As only a small number of *Send Probe* messages are sent, the number of retransmissions and collisions that happen during the probing process are included in the channel decision process as it affects the channel's reliability. As the retransmissions and collisions are a link layer process, the values are kept in a temporary *Retransmit Table* and it is included to be sent in the next *Send Probe* message. This is because the value is only valid for that transmission. It gets reset each time a new packet is sent or received. The table is accessed from the application layer before the next *Send Probe* message is sent. The *Send Probe* message includes the current number of probe message and the number of tries (retransmissions and collisions) the previous packet had taken before it is successfully received. These values are used to decide if the channel is better than the previous channel if it has better probing result, meaning less retransmission.

The node keeps the value of all probing messages it receives. It sends the *Probe Result* message to the LPBR as shown in Figure 4.6(d). Unlike the LPBR, the node has a limited RAM, which resulted in past probing values to be stored in the centralised LPBR. The node however, keeps the probing results temporarily before the final decision of the channel is made. The LPBR could use the information from the node's *Probe Result* to decide on a channel or blacklist bad channels.

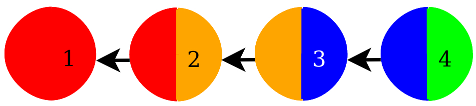
The node then uses the values to decide whether the new channel is better than the previous channel by setting a threshold. The node sends *Confirm Channel* message to all the neighbours to confirm the channel it will be listening on as shown in Figure 4.6(e). The channel can be the new channel or the node can revert to the previous channel depending on the *Probe Result*. The neighbours will send an acknowledgement back to the node confirming the change. This is also important to ensure that all neighbours could communicate with the node on the correct channel. The neighbours will update their neighbour table of the node's channel.

4.3.3 MAC Layer

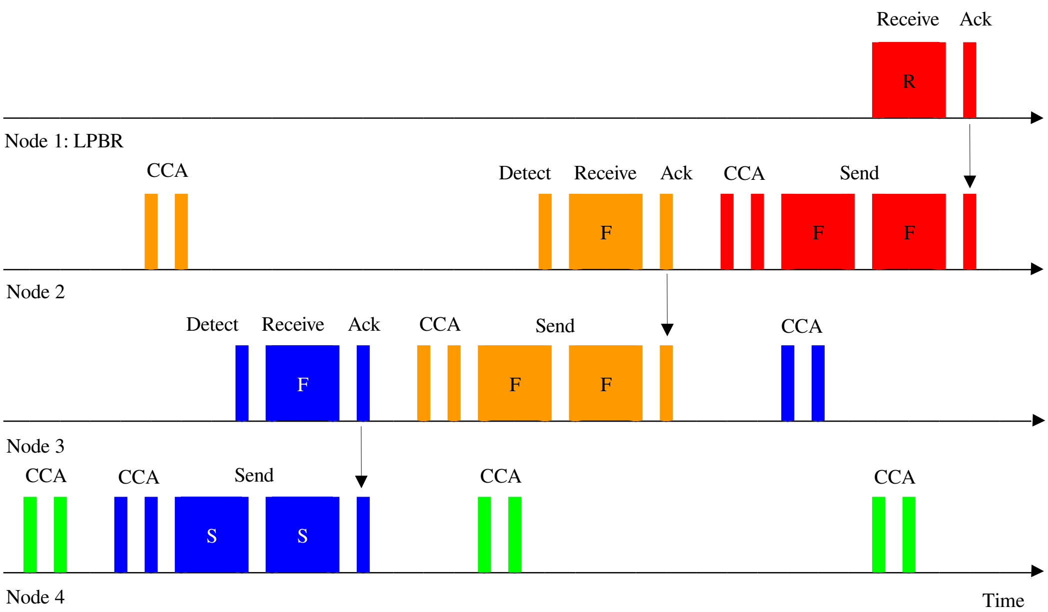
As explained in Section 4.2.2, packets that have not been transmitted are queued in the buffer. The transmitting channel is set at the MAC layer as packets are not sent immediately if there are packets being queued. ContikiMAC is a single channel protocol. It is modified to support multichannel protocol while complying with the same low-power ContikiMAC principle. Each time the node goes to sleep, it will wake up on its listening channel waiting for the incoming packets. If the node has a packet to send, it needs to change to the transmitting channel.

In order for the packet transmission or retransmission to be on the correct channel, the neighbour's channel saved in the *neighbour table* at the network layer is accessed from the MAC layer and the channel is set to the transmitting channel. The node resets the channel to its listening channel after the transmission succeeded and goes back to sleep.

Figure 4.7 shows an example of the multi hops packet transmission in MCRP. The different colours represent different channels. Node 4's channel is represented by green, node 3 is blue, node 2 is yellow and node 1 is red. At each hop, the node changes to the next hop listening channel as shown in Figure 4.7(a) before forwarding the packet. Figure 4.7(b) shows the transmission processes, which includes the wake up and transmission (forwarding) channel change that happens from node 4 (the sender) to node 1, which is the LPBR. *R* represents the receiving packet, *F* is the forwarding packet and *S* is the sending packet.



(a) Topology diagram



(b) Transmission channels

Figure 4.7: MCRP multi hop packet transmission

The example in Figure 4.7 shows node 4 is sending a packet to node 1, the LPBR through node 3 and node 2. Node 4 wakes up and checks for incoming packets on its channel. As it has a packet to be sent to node 1, it checks the next hop channel, which is node 3 and changes the channel to node 3 listening channel (blue). It checks if the channel is clear for transmission using CCA. The CCA relies on the RSSI threshold to detect the radio activity on the channel. The CCA returns a positive value to indicate that the channel is clear and proceed to send the packet to node 3 on node 3 channel. Node 3 detects the packet when it wakes up and receives the packet. Node 3 sends the link layer acknowledgement to node 4 so that node 4 stops sending the packet. Node 4 goes back to sleep and wakes up at the next cycle on its channel (green) to check for any incoming or outgoing message for the node. If there is none, the node goes back to sleep. As node 3 is not the destination, it forwards the packet to node 2 on node 2 channel and node 2 to node 1, the LPBR, which is the destination node. All nodes reset their channels after the transmission and wake up on their own listening channel.

4.3.4 Network Layer

RPL is explained in Section 2.5. The RPL control messages have been modified in the MCRP implementation. Two main changes to the RPL control messages are the DIS (which is sent by a new node to make it possible for a node to require a DIO message from a reachable neighbour) and the DIO (the main source of routing control information) control message. The DIS and DIO control messages are usually sent using broadcast. However, the DIS and DIO support unicast. MCRP sends the RPL DIS control message in broadcast and the DIO in both broadcast and unicast. This allows new nodes to be detected through broadcast.

Figure 4.8 shows the control messages for multichannel protocol. In MCRP, a new node that would like to join an existing tree needs to send the DIS control message to the reachable neighbours. However, as the reachable neighbours could be on different channel than the nodes were initially during start up, the new node needs to send the DIS message on all channels available to be able to find the neighbours. The neighbour that receives the DIS message will reply with a DIO message and a packet that tells the new node of the neighbour node's channel to communicate on. The new node updates the neighbour table and has successfully joined the tree.

If the neighbour does not receive the DIS (the neighbour will send DIO as a reply to the

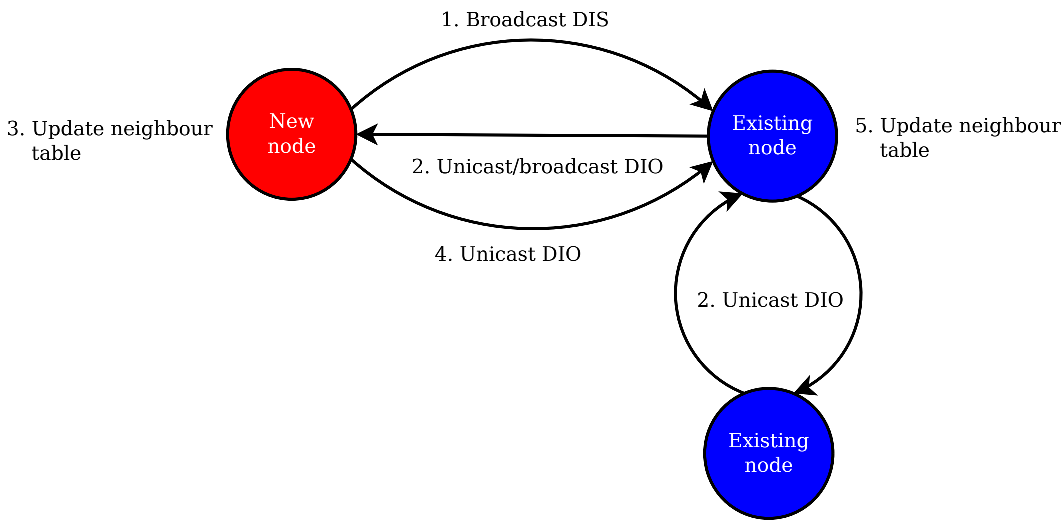


Figure 4.8: MCRP RPL control messages

DIS received) from the new node before it is due to send the DIO message, the neighbour sends a broadcast DIO on the default channel. The new node upon receiving the DIO will join the tree and updates the neighbour table. All neighbours send a DIO broadcast on the default channel and a DIO unicast for known neighbours on channels that the neighbours are listening on.

One of the main reasons for this is because broadcasting on all channels would require more energy and it would take a longer time before all reachable nodes could receive the control message. This could delay changes that might happen in the tree, i.e. changing of parent node. Secondly, all nodes by default will switch on to the same default channel as that is how the nodes are being set up.

4.4 Summary

This chapter explains the specific Contiki components, which are the communication stack, buffer management, routing and network interface that are essential to MCRP. MCRP implementations on cross layers are described in detail concentrating on the application layer, where most of the multichannel decisions are made, the network layer, where multichannel information are stored and retrieved, and the MAC layer to execute the channel changes instructions.

Selecting low interference channel in addition to switching to the correct channel to ensure reliable packet transmissions and receptions is vital as MCRP is an asynchronous protocol. Information on the channel is exchanged during routing formation and periodically through control packets to ensure that the network is connected and reachable. The new nodes are able to join an existing tree through control packet broadcast on all channels including the default channel to detect a nearby node. The new nodes will then communicate using unicast on known channels to neighbours to reduce the number of packets transmitted, thus energy spent.

Chapter 5

Simulation Performance Evaluation

5.1 Introduction

This chapter presents the evaluation of MCRP. The experiment setup for Cooja simulation is explained below. MCRP is evaluated using an end-to-end packet delivery performance metric. The results from the experiments are presented and discussed.

5.2 Experimental Setup

MCRP is evaluated in the Cooja simulated environment. In Cooja simulation, an interference model is used, as simulation allows full control over the test environment and the experiments are repeatable. Although the interference model does not fully mimic the behaviour of the real-world interference, it enables MCRP performance to be tested in various conditions when the channel performance is degraded and therefore gives a better understanding of the performance.

5.2.1 Simulation

MCRP is evaluated in the Cooja simulated environment with emulation of TelosB nodes that feature the CC2420 transceiver, a 802.15.4 radio. The nodes run on IPv6, using UDP with standard RPL and 6LoWPAN protocols. Figure 5.1 shows the layout of the network. The network consists of 31 nodes that are used to run the simulation, where one node is used as the border router (LPBR) node, 16 are interference nodes, and 14 are duty cycled nodes that act as UDP clients to send packets to the LPBR, spanning over 20-30 metres between each node. The RPL border router is used as the LPBR in order to move most processing decisions onto a PC, as it has more RAM and better processing capabilities than a sensor. A TelosB sensor has limited RAM and ROM of 10K bytes and 48K bytes of flash memory [110]. By using a border router, this allows channel changing to be decided in real-time

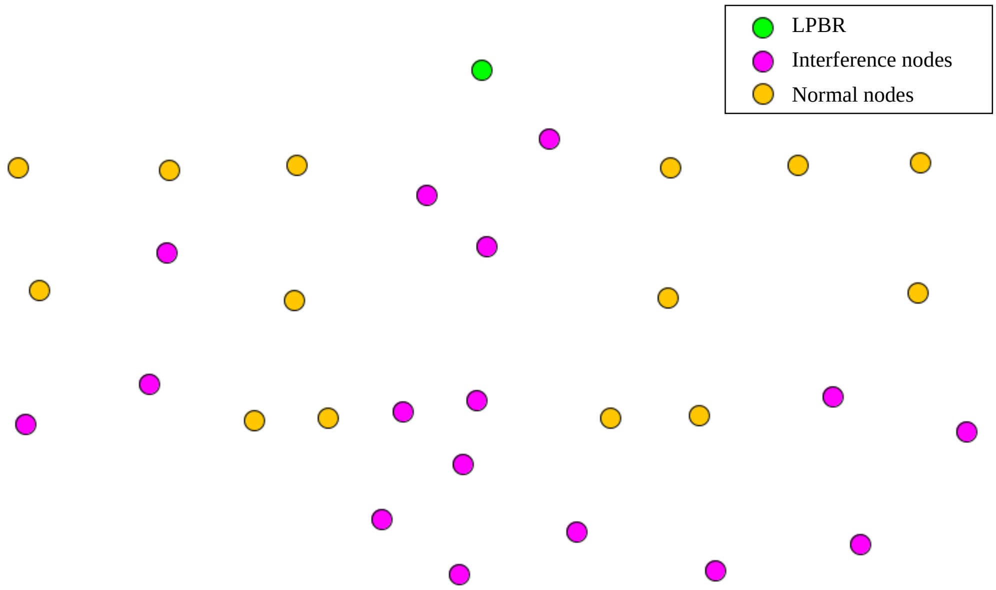


Figure 5.1: Simulation: Layout of the simulation nodes

without draining the memory and battery on a sensor. The border router also acts as the root of the tree.

A controlled interference node that generates semi-periodic bursty interference is simulated to resemble a simplified Wi-Fi or Bluetooth transmitter on several channels at random. The interference model proposed in [14] is used in the simulation to generate a similar packet loss rate to the values of the theoretical and real nodes given in [13]. The model is illustrated in Figure 5.2.

The interference has two states, a clear state (C) and an interference state (I). In the interference state, the interference node generates packets for a time that is uniformly distributed between $9/16$ seconds and $15/16$ seconds. In the clear state the interferer produces no packet and stays in this state for between $3/4 * clear\_time$ and $5/4 * clear\_time$, where *clear_time* refers to the rate of interference (ir) in percentage to represent the interference level. The level of interference used in terms of the *clear_time* is 100% for no interference, 75% for mild, 50% for moderate and 25% for extreme interference. The percentage represents the ratio of the time that the channel is clear for transmission. This means that in the extreme interference case (25%), the interference happens between 0.14 to 0.23 seconds, with clear time of between 0.18 to 0.31 seconds for transmissions.

Interference on multiple channels is used in the simulation to show the hypothesis that MCRP can help avoid interference. The scenario that is considered is where ContikiMAC

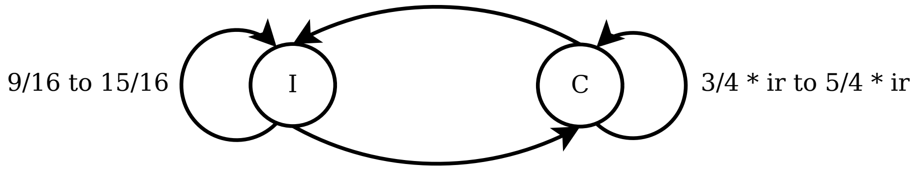


Figure 5.2: Simulation: Interference model

with the RPL system is subject to interference on its channel after set up has successfully completed, so the RPL set up is allowed to complete before interference begins.

The protocol's performance in loss over time in the presence of interference is observed. Two scenarios with interference on channels are considered, where the interference nodes are on different channels: (i) 8 nodes with extreme interference rate, 8 nodes with no interference and (ii) 4 nodes with extreme, 4 nodes with moderate, 4 nodes with mild and 4 nodes with no interference rate.

The interference channels are randomly chosen from the available 16 channels and the same interference channels and rates are used throughout the experiments. However, channel 26 is kept clear from interference in order to ensure the RPL set up is unaffected. In scenario 1, the interference rates are fixed to extreme and no interference, to observe the effect it has on the channel changing decisions. In scenario 2, the interference rates vary, to observe how MCRP copes with deciding a channel when there is more interference than scenario 1 but with less interference intensity.

The simulation runs for a duration of 45-60 minutes to send 210-560 packets. When the nodes are switched on for the first time, all nodes are initialised to channel 26, the default channel for Contiki MAC layer. RPL is allowed five minutes to set up (which is ample time). The RPL topology is formed in a minute. The simulation waits for another five minutes to allow the trickle timer to double the interval length so that the RPL control messages are not being sent frequently. The multichannel protocol is then run for 25 minutes. In the simulation with 15 nodes, the protocol takes 20-25 minutes to run the channel change set up. Another 5 minutes is allowed if retransmissions happen. In a single channel protocol simulation, all the nodes are changed to channel 22 after 5 minutes of the RPL set up time. This allows RPL to have enough time to discover all nodes, to form an optimised topology. The topology formation does not form completely if the interference nodes interfere from the beginning.

The interference node starts sending packets to interfere after 3 minutes the system is switched on, so that the interference channel is involved in the channel changes decision. It shows that the protocol tries to avoid changing to the interference channel through time out and probing failures. After 30 minutes, the client nodes will send a normal packet periodically every 30-60 seconds to the LPBR. This is done in order to avoid collision of the nodes sending packets at the same time.

5.3 Results

The performance of MCRP is compared against the standard ContikiMAC with RPL and Orchestra. Orchestra does not support RPL downwards routing due to the limited memory in TelosB. It however, support the upwards traffic, which is required in the experiments as all traffic is directed upwards towards the LPBR.

MCRP is analysed using an end-to-end packet delivery performance metric. The transmission success rate is calculated from the sender to the receiver over multiple hops.

The simulations are repeated ten times. In all plots, the mean value of the ten simulations is plotted with error bars corresponding to one standard deviation in either deviation to give a measure of repeatability. The plots are of the proportion of received packets (from 0% to 100%) against time, where the loss is measured over the previous time period. The x-value is shifted slightly left and right to prevent error bars overlapping.

5.3.1 Packet Loss Rates

The performance obtained in ContikiMAC with RPL (single channel protocol) is compared with MCRP in terms of packet loss rate. As described previously, levels of interference used (referred to as *clear_time* in [14]) vary among 100% (no interference), 75% (mild), 50% (moderate) and 25% (extreme) where the percentage is the ratio of the time the channel is clear for transmission. All of the tests have a common format: RPL procedure is allowed to set up without interference in order not to bias subsequent tests. Then the interferers begin to operate with a constant level (none, mild, moderate or extreme).

5.3.1.1 Simulation

Figure 5.3 shows the results in simulation for ContikiMAC with the RPL protocol. It can be seen that the level of packet loss varies considerably between experiments (the error bars are always large). It can also be seen that even for mild interference there is considerable loss and this gets worse as time proceeds. In the extreme interference case the loss always goes

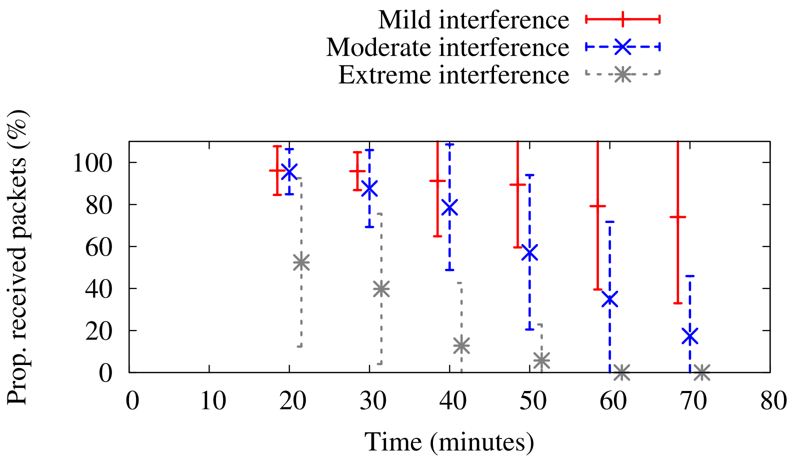
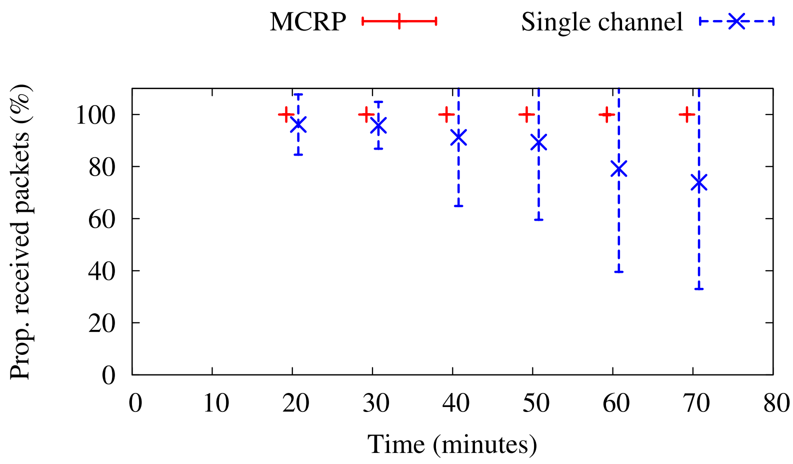


Figure 5.3: Simulation: Level of packet loss for mild, moderate and extreme interference levels using single channel protocol

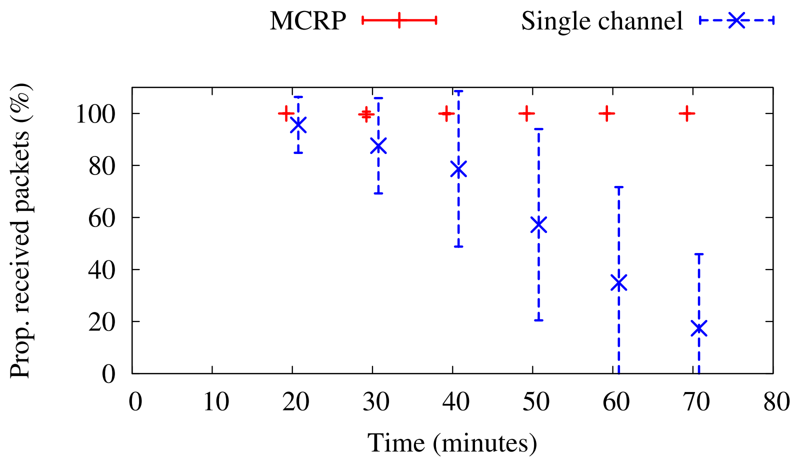
up until no packets are received. For mild interference the system evolves until it is losing around 20% of packets but the number of packet loss increases over times. This shows that in the presence of interference, single channel protocol shows deteriorate reception rate. The reason for these losses is because the network is being congested with retransmissions packets.

The results from the single channel protocol with interference is compared with the multichannel protocol with the same interference rate of 75% (mild), 50% (moderate) and 25% (extreme). The test is done to evaluate MCRP behaviour in different interference rate and to compare the result with a single channel protocol case.

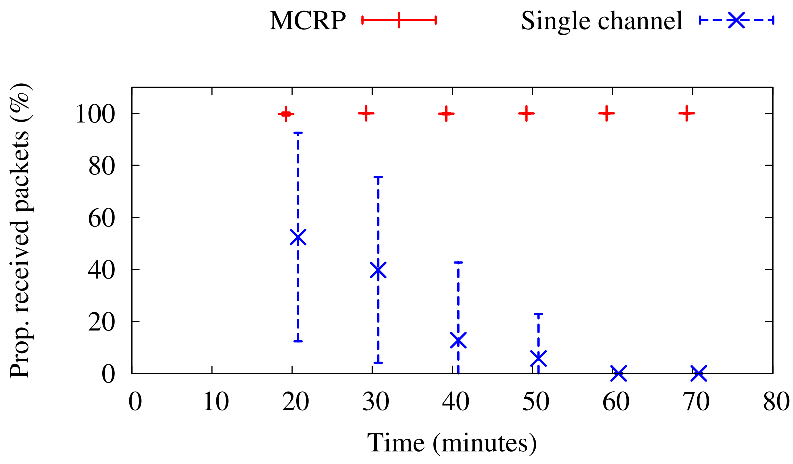
Figure 5.4 shows the average results from ten runs that were done. It can be observed that during high and moderate interference, if the LPBR tries to send a channel change value that is the same channel as the interference, the request will either time out or if it succeeds, the probing messages received are less than a threshold that allows the node to change its listening channel to the new channel. This is as expected as MCRP checks the channel each time before deciding on the new channel to avoid interference channel. By doing this, it ensures that the node's listening channel is a good channel. This enables the use of all available channels without blacklisting any channel until it is sure that it is a bad channel through the probing process. The channel quality table is built at the LPBR that over time can be used to learn good and bad channels based on several probing processes. MCRP avoids the interference channel, which resulted in less loss than in a single channel protocol case.



(a) Mild Interference



(b) Moderate Interference



(c) Extreme Interference

Figure 5.4: Simulation: Results of MCRP and a single channel RPL protocol on different interference rate

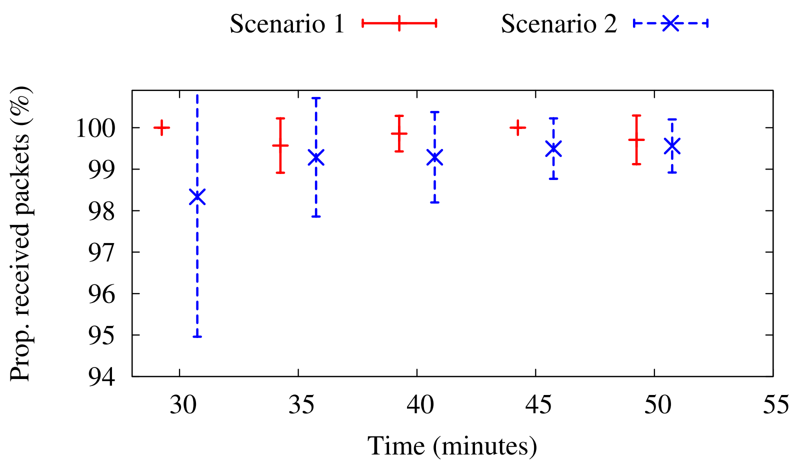


Figure 5.5: Simulation: Level of packet loss for scenario 1 and scenario 2 using multichannel protocol

In the single channel protocol, the node does not have enough time to recover from the interference to retransmit and drops all packets. Figure 5.4(c) shows that there are more packets being dropped over time and the node stops receiving packets as it does not have enough buffer space to store the incoming packet and the channel becomes congested. However, as the interference rate increases (less interference), the single channel protocol performance improves as it has more time to recover.

In the mild interference case shown in Figure 5.4(a), all probing messages for MCRP are received even though there is interference in that channel. This means that the channel can be used for transmission. The interference does not heavily affect the transmissions in a single channel protocol in the beginning as the interference is not frequent enough, showing 60% to 95% reception rate in the first 30 minutes. The node has enough time to recover from the interference through retransmissions. However, the interference would affect the packet transmission over time as the reception rate dropped to 40%. MCRP, on the other hand, kept high reception rate of near 100% throughout the simulation period.

To further evaluate MCRP capabilities to cope with interference from many sources, thus channels, two interference scenarios are considered. In scenario 1 half of the channels (including the original channel) have no interference at all and half the channels have extreme interference. In scenario 2, four channels (including the original channel) have no interference, four have mild, four moderate and four extreme interference. Figure 5.5 shows multichannel protocol results for these two scenarios. In scenario 1 the protocol performs extremely well, the packet loss is near zero and the protocol successfully detects channels

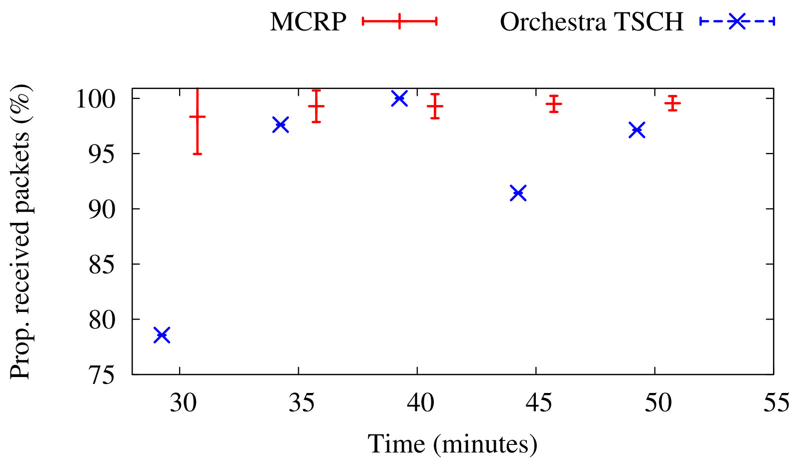


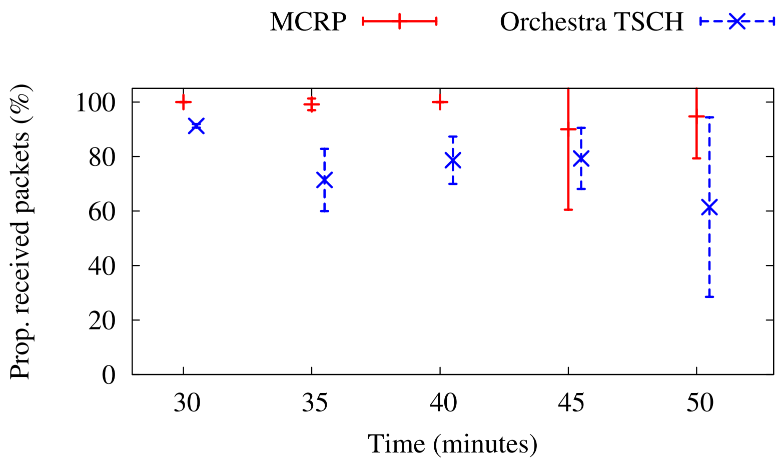
Figure 5.6: Simulation: Level of packet loss on testbed for MCRP and Orchestra

with interference. Scenario 2 has similar results as in scenario 1. The protocol does well at reducing the effects of interference and could detect moderate and mild interference. Scenario 2 shows variation of reception rate, which reduced over time. This is because as there is less number of channels with no interference, some of the nodes are using the channels with mild interference. The nodes could recover from mild interference and retransmit packets as the other nodes are on the other channels, thus the nodes do not compete for transmissions on the channel.

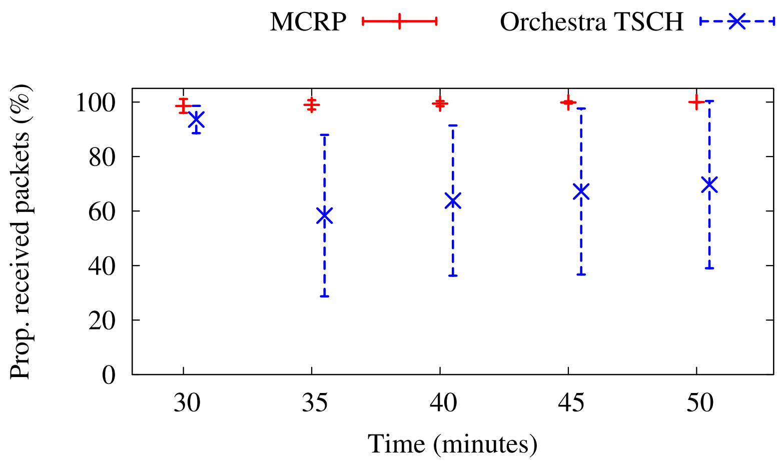
5.3.1.2 MCRP vs Orchestra

To evaluate MCRP capabilities to cope with interference from many sources, thus channels, and to prove that MCRP performs better than not only a single channel protocol, MCRP is compared to an existing multichannel protocol Orchestra. The emulations were run, where (i) the layout and nodes' interference channels are fixed as shown in Figure 5.1, (ii) the layout is fixed while the interference channels are chosen by random and (iii) all the nodes including the interference nodes are placed randomly. In the experiments, Orchestra uses channel hopping on all 16 channels.

Figure 5.6 shows the result from scenario 2 on both MCRP and Orchestra in the fixed layout. Orchestra has low packet loss showing around 90-100% received packets as it hops on all channels, which includes the channels that have higher interference. In comparison, MCRP selects certain channels to change into after checking the channels' condition, which gives MCRP nearly zero packet loss. Orchestra shows good result as it hops to another channel in the next iteration, which allows it to move from the interference channel faster to be able to keep the loss rate to a minimum. While Orchestra has high proportion of received



(a) Fixed layout



(b) Random layout

Figure 5.7: Simulation: Level of packet loss for MCRP and Orchestra

packets, MCRP shows near 100% packet reception. Orchestra shows no deviation as the channel values are fixed for each iteration thus giving the same results each time unlike MCRP, where the channels are selected at random before it is used.

Figure 5.7 shows MCRP and Orchestra results for the fixed (with random interference nodes) and random nodes layouts. MCRP performs extremely well in both scenarios as the average packet reception rates are between 90%-100% and the protocol successfully detects the channels with interference. Orchestra has higher packet loss compared to MCRP, showing a maximum of 40% packet loss on average as the channels with interference are being used for transmission periodically. In comparison, MCRP selects certain channels to change into after checking the channels' condition, which gives MCRP a smaller number of packet loss. MCRP avoids the interference channel while Orchestra hops to the next channel in the next iteration for transmission.

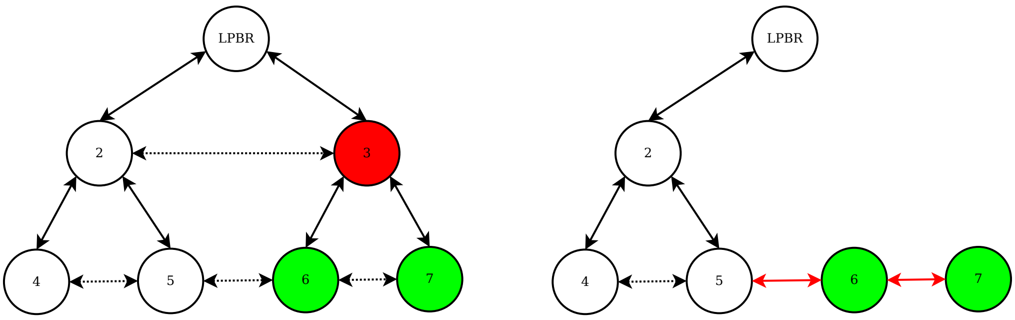
In MiCMAC [5], it is stated that MiCMAC has a transmission success rate of 99% when using four channels. However, when more than four channels are used (8 or 16 channels), MiCMAC performance degrades to approximately 88% (16 channels) due to interference channels. The interference model that MiCMAC uses is different than in this experiment. They compared the result with Chrysso, where Chrysso has a transmission success rate of approximately 88% for 4 and 8 channels and suffers greatly in the case of 16 channels with 60% success rate. MCRP, on the other hand, shows greatly reduced loss rate with any number of channels at approximately 99%.

5.3.2 Setup Overhead

Obviously the system of changing channels and probing to see if a channel is free of interference introduces a certain amount of overhead into the protocol. This takes the form of (a) extra messages passed and (b) extra time taken to set up. Default RPL on ContikiMAC for the topology considered in these experiments completed its set up using 276 packets. MCRP, the multi-channel protocol completed its set up in 716 packets, that is an overhead of 440 packets on top of RPL.

This overhead comes from the channel changing messages to nodes and neighbours, probing messages, channel confirmation messages and acknowledgement packets, which are required to ensure a thorough channel change decision. However, it is worth mentioning that this is a one-off cost. This represents (in this experimental set up) approximately one hour of extra packets in the situation of a deployment that is meant to work for weeks or months. Typically, the interference channels at a location (offices) are the same for a period of time with slight changes due to new devices registered. As this does not happen frequently, MCRP does not need to run periodically. However, MCRP can be invoked dynamically when the current channel deteriorate due to interference. The reconfiguration overhead is negligible in the context of more successful packet transmission for a longer period of time compared to the overhead packets based on the typically unchanged interference patterns as shown in the real-world office environment channel occupancy over a duration of one week [97].

In terms of set up time, the protocol begins to change channels only when the RPL set up process is complete (or at least stabilises). The set up time is 1154 seconds beyond the RPL set up time of 286 seconds. However, it should be noted that, in fact, the system



(a) Initial tree (b) Reconstructed tree when node 3 is removed

Figure 5.8: Simulation: A simple simulation layout to test the tree reconnection

remains fully functional and capable of sending packets during the set up so this set up overhead does not matter to data transmission. Therefore it can be concluded that data sending costs (extra packets) of set up are negligible in the context of a deployment that will last more than a day. The extra set up time is also negligible within this context and furthermore does not degrade the performance of the network during this set up phase.

5.3.3 Channel Switching Delay

Each node has a different listening and transmitting channels. When the node is awake, it waits for the incoming packets on its listening channel. If the node has a packet to send, it will switch to the next hop listening channel based on the channel information from the neighbour table. The channel switching takes at most $100\mu s$ to switch to the transmission channel. This delay is negligible in the low packet rate WSN. MCRP ContikiMAC uses a transmission phase-lock, where the transmission node knows the receiver wake up phase. The node starts transmitting just before the receiver is expected to be awake. The channel switching happens shortly before the receiver is ready to receive the packets, thus the time taken in channel switching does not affect the packet reception. The node goes back to sleep once the transmission has succeeded or reached the maximum number of retransmissions (packet loss). In the next iteration, the node is reset and wakes up on its listening channel.

The channel reset is done in these cases: (i) the queue buffer is empty, (ii) before sending the next packet from the queue buffer, and (iii) the last packet in the queue buffer has been sent. This reset is done to avoid any delay in packet reception that could happen when the node is awake.

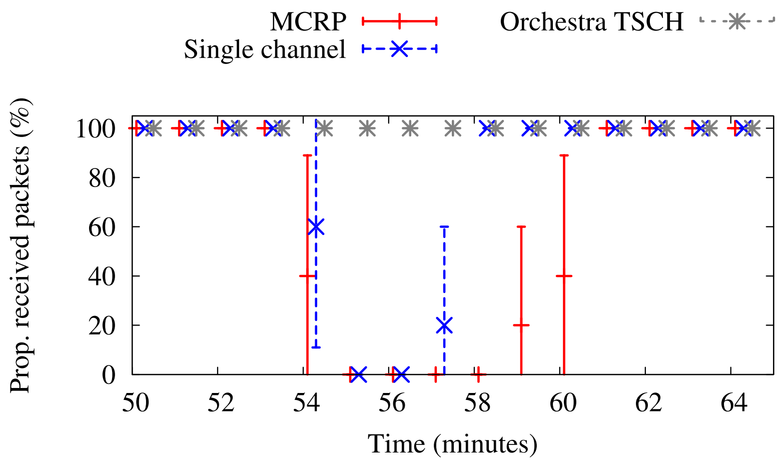


Figure 5.9: Simulation: Reconnection time taken for MCRP, single channel protocol and Orchestra

5.3.4 MCRP Reconnection

Figure 5.8 shows the experimental setup where there is no external interference to get an accurate convergence time of the topology. The dotted lines represent potential paths and the solid lines are the selected paths. Node 3 is disabled after 53-54 minutes (25 packets are sent and received). Node 6 and 7 route through node 3 to the LPBR. When node 3 is dropped, node 6 and 7 have to find another route, which is through node 5 and 2 to get to the LPBR. The time taken for the nodes to reconfigure the routes and the number of packet loss are showed in Figure 5.9.

In MCRP, it took between 5-7 minutes before node 6 and 7 discovered and reconnected with the tree to proceed with the transmission. Single channel protocol however, was slightly quicker, taken 3-5 minutes. The reason for this is MCRP control packets are sent on several channels, thus it would take slightly longer to be able to reach all nearby nodes that might be on different listening channels. A single channel protocol, on the other hand, could send a broadcast to the nodes, which help to reduce the time taken during the topology reconnection. Local caching of the neighbour nodes' channel information could speed up the convergence time. However, this could overflow the node's memory as the node has to cache all the neighbours' information.

Comparing to Orchestra, Orchestra is a synchronous protocol. It has a dedicated slot and periodic schedule for the RPL signalling, which means it detects the failed node quicker unlike in MCRP and the default single channel asynchronous protocol. The results from the Orchestra simulation shows that as Orchestra has a slot checking the nodes every minute, it is able to reconnect the nodes without having any packet loss. The disadvantage of Orchestra is, the nodes are listening on the same channel during the broadcast. As the channel is fixed and known to all nodes, it is more prone to attack. Also, even though Orchestra introduces priority to the traffic, the RPL traffic is sent frequently at every period if there is no other higher priority traffic. The trickle timer that is used by the default RPL has the advantage of reducing the number of redundant control packets by doubling the waiting time for the control packets. Orchestra detects failed node quicker at the cost of frequent control packets that are redundant in a stable topology, which increases the use of bandwidth and nodes' energy consumption.

5.4 Summary

This chapter demonstrates MCRP abilities in dealing with external and internal interference. MCRP is tested in the simulated environment to study the effect of multichannel protocol to the performance in a controlled environment. MCRP results are compared to the standard single channel protocol, ContikiMAC and the multichannel protocol, Orchestra that implemented TSCH in terms of the end-to-end packet delivery. The setup overhead, the channel switching and the reconnection delay in MCRP are discussed, to demonstrate that these values are negligible in the context of WSNs that could run in years while maintaining high throughput.

Chapter 6

Testbed Performance Evaluation

6.1 Introduction

This chapter presents the performance evaluation of MCRP on our own testbed (referred to as the MCRP testbed). The experiment setup for the MCRP testbed is explained below. Similar to the simulation, MCRP is evaluated using an end-to-end packet delivery performance metric. The results are presented and analysed.

6.2 Experimental Setup

The MCRP testbed experiments provide the ability to validate MCRP performance in real wireless channel environments, unlike simulation. However, the network's behaviours are complicated to examine, as the experiments are not repeatable. The environmental condition and the network could be different at each iteration depending on the location, time and channels' occupancy. The authors in [97] compared the channel occupancy in the office and home environments, which potentially can have distinct channel usage. This affects the results differently, as the channel conditions could have drastic changes during the run time in the residential environments more than in the office environments.

6.2.1 Channels' Occupancy

Figure 6.1 shows the occupancy of different channels in three different locations: residential, public area (cafe) and university (Malet Place Engineering Building, University College London). It can be seen that most channels are occupied in the residential environment and there is a limited number of available channels in the university environment that are free from interference. This shows that it is extremely difficult to find a good interference-free channel and that it varies from one location to another. The experiments of MCRP on the MCRP testbed took place in both residential and university environments with the same

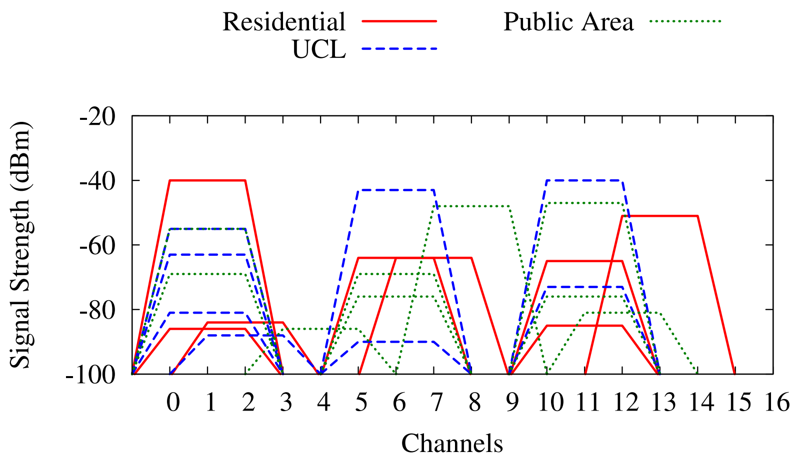


Figure 6.1: Real world: Interference level on the channels at different locations

Power Level	Output Power (dBm)	Current Consumption (mA)
31	0	17.4
27	-1	16.5
23	-3	15.2
19	-5	13.9
15	-7	12.5
11	-10	11.2
7	-15	9.9
3	-25	8.5

Table 6.1: Output power configuration for sensor CC2420 radio

experimental setup. The MCRP testbed results are compared to the single channel protocol case, to analyse MCRP performance in various environments.

6.2.2 Output Power Level

The sensor uses the Chipcon CC2420 radio that is IEEE 802.15.4 compliant for wireless communications. Power level refers to the sensor's CC2420 radio programmable output power, where the default power level 31 corresponds to the output power of $0dBm$ and current consumption of $17.4mA$. The sensor datasheet [104] lists the values of the power levels and the corresponding values, as shown in Table 6.1. Power level 1 and 2 have the output power of approximately $-50dBm$ and $-40dBm$.

A small number of nodes is used in order to confirm that MCRP is working. The network consists of 7 nodes: 1 border router and 6 duty cycled nodes. The nodes are placed within at least 1 node's range at the power level of 2, which should have nearly 100% packet reception given that there is no interference at the range of around 20 cm.

It is done to have a smaller scale network where all nodes would have the same inter-

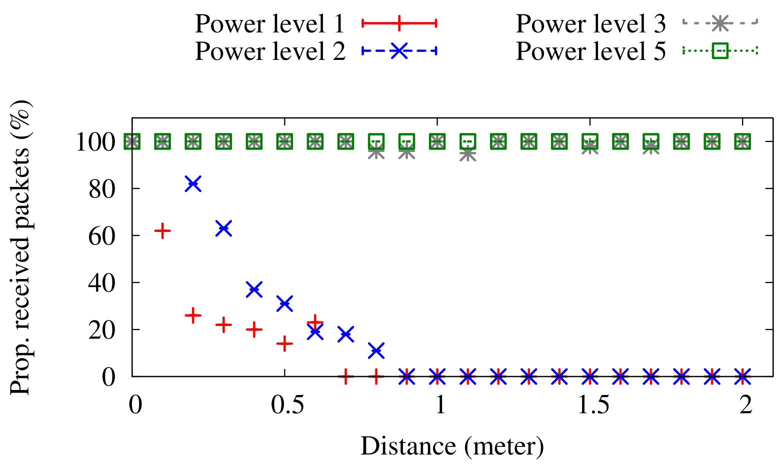


Figure 6.2: Real world: Packet transmission range for different power level

ference source that affects the nodes. Also, to ensure that the nodes would have at least one hop to the sink to fulfil MCRP criteria in changing channel processes. This experiment can be duplicated to cover a larger scale, as the radio has the range that could span over 20-30 metres.

The authors in [71] studied the distance factor in TelosB. Their results are checked by running a similar experiment to observe and confirm the relationship between the power level, distance and the ratio of successful packet transmission. In the experiment, 2 sensors are used, where one is the sender and one is the receiver. The distance between the sensors is increased until there is packet loss, before testing with the other power levels. TelosB has the power level ranging from 0 to 31 levels [104]. These transmission power levels can be changed at compile and run time. The proportion of received packets of different power levels and the distance is shown in Figure 6.2.

Power level 1 shows a high reception rate when the distance is less than 0.2 m, power level 2 is less than 0.3 m, power level 3 shows minor packet loss and power level 5 shows no packet loss at 2 m. All the packets are received successfully when the distance between the sender and receiver is less than the steadily-transmitted distance that corresponds to the power level.

6.2.3 MCRP Testbed

The MCRP experiment is run for a duration of 2 hours to send 300 packets, which is 50 packets per node, sending 1 packet per minute. As the nodes are switched on at nearly the same time, RPL is allowed five minutes to set up. MCRP is run for 45 minutes to allow for the process of channel changes. The nodes wait for the MCRP process timer to time out

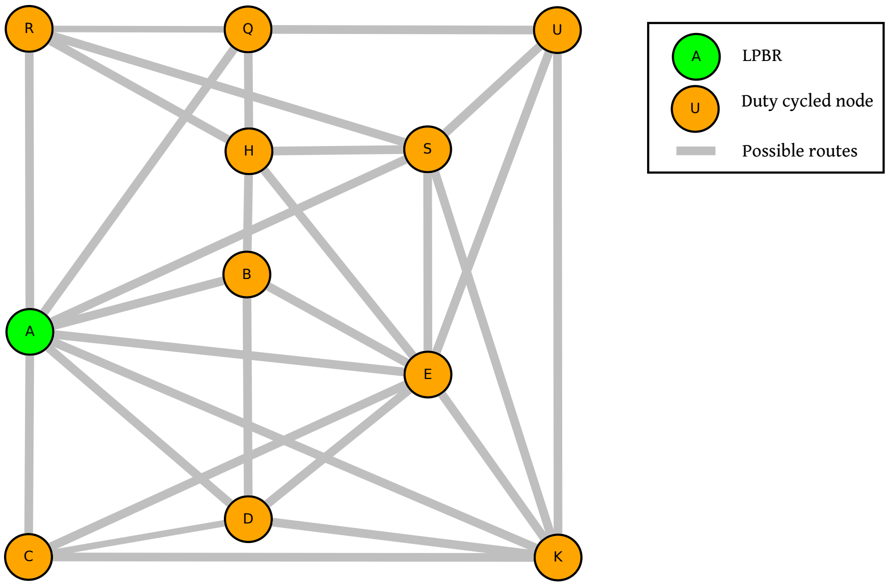


Figure 6.3: MCRP testbed: The layout of the nodes

before the nodes can send normal packets. The experiment is then repeated with 11 nodes (1 LPBR, 10 duty cycled nodes) to study the effect that MCRP has towards the increased number of nodes. Figure 6.3 shows the layout of the nodes and the possible routes that the nodes could choose to get to the LPBR.

The environment condition and the network could be different at each iteration depending on the location, time and channels' occupancy. The authors in [97] compared the channel occupancy in the office and home environments, which potentially can have distinct channel usage. This affects the results differently as the channel conditions could have drastic changes during the run time in the residential environment than in the office environment.

The experiment of 11 nodes network (the same layout as previously) is repeated on different environments. The experiments of MCRP were taken place in both residential and university environments with the same experimental setup. The results are compared to the single channel protocol case to analyse MCRP performance in various environments.

6.3 Results

MCRP is compared against the single channel ContikiMAC with the RPL protocol. Similar to the simulation, the end-to-end packet delivery is used as the performance metric. The proportion of received packets from the sender over multiple hops is plotted with the er-

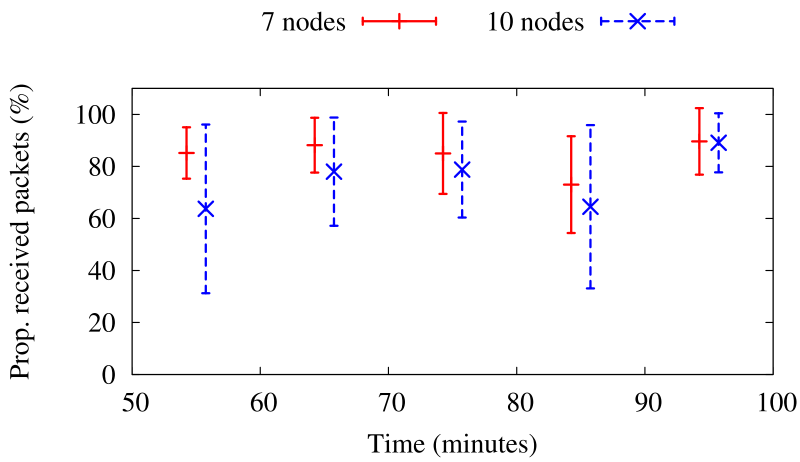


Figure 6.4: MCRP testbed: Level of packet loss for MCRP in real-world environment

ror bars corresponding to one standard deviation in either deviation to give a measure of repeatability. The values on the x-axis are shifted slightly to avoid overlapping error bars. The experiments of the testbed are repeated ten times.

6.3.1 Packet Loss Rates

As mentioned previously, the interference could occupy and affect the channels differently at each run. Unlike in the simulation, the RPL tree formation set up is affected by the interference during initialisation. As a result, the network could be formed differently at each iteration.

In our own MCRP testbed, the tree topology was formed differently each time depending on the radio coverage and interference level. This affects the RPL ETX value for the next hop selection. Figure 6.4 shows the result from the experiment. It can be seen that the number of received packets vary from approximately 50% to nearly 100% with better results when using a smaller number of nodes, 7 nodes than 10 nodes. The reason for this is because in smaller number of nodes, each node has less number of neighbours, thus avoiding interfering with each other (if the nodes were transmitting at the same time to the same receiving node). However, in both results, the number of packets lost decreases over time.

Figure 6.5 shows the result of MCRP compared to a single channel protocol in mild to moderate and extreme interference. The single channel protocol with mild interference has the signal strength approximately $-65dBm$ while the extreme interference is $-40dBm$. These channels are used to interfere with the transmissions. Referring to the interference graph in Figure 6.1, it can be seen that most channels are occupied except for channel 26 (labelled as 16), which means, MCRP could hardly find a clear channel for transmission

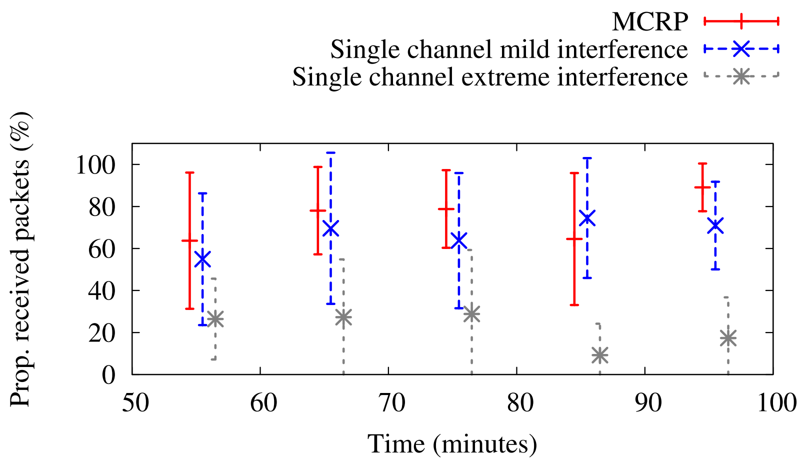


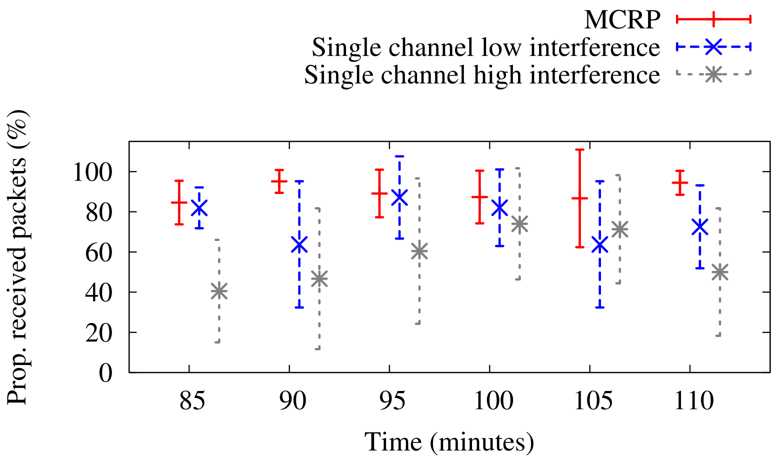
Figure 6.5: MCRP testbed: Level of packet loss for MCRP and single channel protocol

thus the proportion of received packets to be around 50% to 100% unlike the simulation results that show high reception values. In the single channel protocol case, it can be seen that the results are acceptable in mild to moderate interference case. However, it shows low packet reception rate in the extreme interference case. This shows that MCRP has more advantage than a single channel protocol.

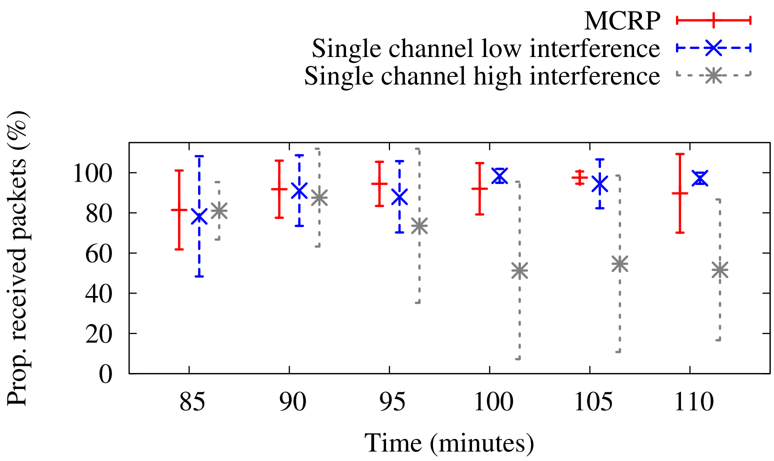
Comparing MCRP to the single channel protocol result, MCRP shows promising result over time. It requires more experiments to be undertaken with some changes to MCRP to run channel changes periodically in order to ensure that it could provide a better number of received packets with small standard deviation than it currently is for higher reliability.

Figure 6.6 show the results from the experiment in residential and office environments for MCRP compared to a single channel protocol in low and high interference. It can be seen that in both graphs, the number of MCRP received packets vary from approximately 80% to 90%. This is because there is a limited number of clear channels that are not affected by interference unlike in the simulation, where the interference is introduced.

In the single channel protocol case, it can be seen that the results are acceptable in the low interference case. It shows better result in the office environment than in the residential environment, as the spectrum usage in the office environment is typically centrally managed. However, both graphs show smaller number of packet reception rate in the extreme interference case compared to the other cases. This shows that MCRP has more advantage than a single channel protocol in extreme interference regardless of the locations. The results show that multichannel protocol performed better than a single channel protocol in dealing with the unpredicted interference occupancy in the 2.4 GHz frequency band.



(a) Residential environment



(b) Office environment (UCL)

Figure 6.6: MCRP testbed: Level of packet loss for MCRP and single channel protocol in different environments

6.4 Summary

This chapter presents MCRP results in the real-world environment. Different than the simulation, the interference level and channels' occupancy vary depending on the location. Most of the channels are occupied, which makes MCRP appealing than a single channel protocol, as it could use several channels for transmission rather than a single channel that could have high interference over time. MCRP shows promising results in the unpredicted interference occupancy in the 2.4 GHz frequency band with high packet reception rate of over 80%.

Chapter 7

Energy Efficient WSNs

7.1 Introduction

There have been various studies to estimate the node's energy consumption in real-time. There are two main ways that are usually studied for an energy-efficient WSN, which are through MAC and routing protocols. In MAC protocols, the radio duty cycle is exploited to minimise the radio usage, which, as a result, enables the nodes to be awake efficiently for transmission or reception without wasting energy idling. In terms of the routing protocols, the nodes' load has to be fairly spread to use different nodes during communication. This is because nodes that are closer to the sink are the most constrained. Those nodes have more traffic to forward, which resulted in more bandwidth and energy being used than the other nodes.

Another important factor that effects the energy consumption that has been extensively studied through MCRP processes in this thesis is the condition of the radio link. By using a reliable radio channel, retransmission could be avoided. Multichannel protocol not only could reduce the end-to-end delay, but it also helps to improve the nodes' energy efficiency by ensuring minimal packet retransmissions, thus reducing energy consumption.

Most solutions that estimate the nodes' energy consumption use metrics such as the radio duty cycle and end-to-end delay to represent the energy consumption. One of the reasons for this is that the estimation of the energy consumption is often used to compare different nodes, therefore the voltage is not required to be computed. It is possible to measure the battery level for the battery-powered sensors. However, it cannot be directly translated for energy estimation because the voltage level of the battery does not linearly translate to the amount of remaining lifetime.

This chapter describes the implementation of a Contiki existing energy module. It also shows the estimation of the energy consumption computation and the implementation in MCRP. The energy consumption in MCRP is evaluated in terms of the energy taken in packet transmission and packet forwarding. The results showed that MCRP consumes less energy than a single channel protocol with interference.

7.2 Contiki Powertrace

Contiki has an energy estimation module called *Energest* that is used to keep track of the per-component energy or power consumption in real-time for the radio during transmit, receive, low power and full power modes. *Energest* measures the estimated energy by using the time that the radio starts and stops in the particular state.

The Powertrace system [36] is a software-based power state tracking system that is able to profile the power behaviour at the network level for WSNs. It computes the estimation of the power consumption at run time per-activity power cost, such as the different states for wake ups, transmissions and receptions of a node. The Energest power state tracking profile (Powertrace) shows an accuracy of within 94% compared to the oscilloscope energy measurements. Powertrace uses the Contiki energy estimation module (Energest) to deduce the period that the sensor node is in the specific state with regards to the radio activity to calculate the average energy or power consumption.

Powertrace uses power state tracking to estimate the total power consumption from the individual's energy capsule at the node level activities. Each energy capsule represents an activity that contains a set of power states such as radio transmission, radio listening, CPU active and CPU sleep. The energy capsules can track the energy at any given time over different states by opening and closing the capsules. Examples of energy capsules are a *wake-up capsule* that has the state of radio listening and when the CPU is active, and a *transmission capsule* that has the state of radio transmission, radio listening and CPU active, as those are the states that contribute to the capsule's energy consumption. These individual energy capsules are the attributes to the network level activities.

Figure 7.1 shows an example of the Powertrace power profile. The Powertrace first tracks the system's periodic wake up before performing a packet transmission *(wake up capsule)*. When a packet is being transmitted, Powertrace tracks the power states and records the estimated energy of the system in the energy capsule, which in this case is the *transmission capsule*. Applications or protocol activities such as routing, forwarding and control traffic require several energy capsules to keep track of the estimation of the system's energy

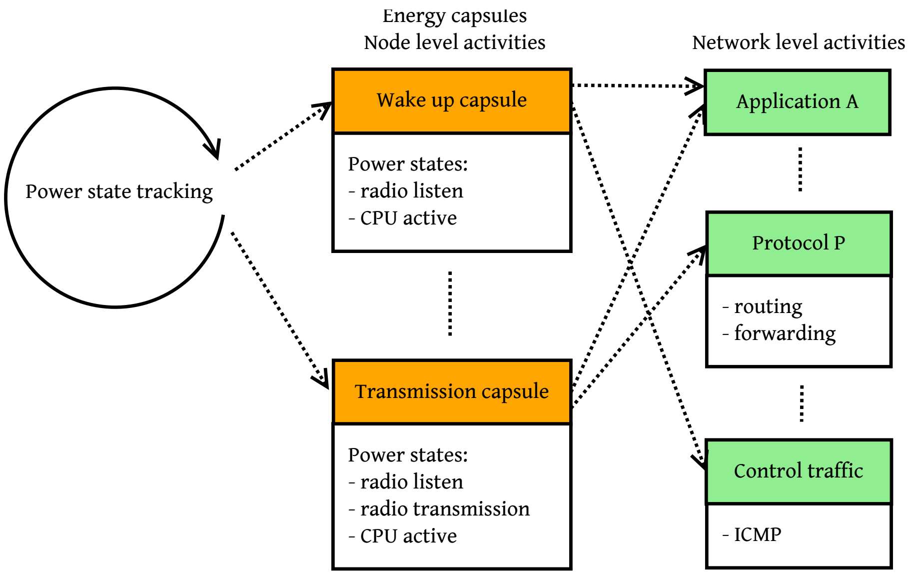


Figure 7.1: Powertrace power state tracking using energy capsules to estimate the system energy consumption

consumption. As an example, a node that is forwarding a packet to another node is required to perform a periodic wake up before attempting packet transmission. The energy consumption of the individual activities is recorded in the energy capsules (the wake up capsule and the transmission capsule).

7.3 Real Time Energy Estimation

Powertrace uses the software-based on-line energy estimation mechanism [35] to estimate the node's current energy consumption in real-time. The on-line energy estimation is implemented in Contiki OS. The energy estimation module uses time measurements that can be directly obtained from the microprocessor on-chip timer when the component is switched on to produce a time stamp. The time difference from when the component was on and when it was later switched off is computed. The current draw of the component listed in the TelosB data sheet is used to compute the total energy consumption estimation, E.

$$\frac{E}{V} = I_m t_m + I_l t_l + I_{tx} t_{tx} + I_r t_r + \sum_i I_{c_i} t_{c_i} \tag{7.1}$$

Equation 7.1 shows the energy consumption model [35], E where V is the supply voltage, I is the current draw and t is the active time computed in Powertrace for m the microprocessor, l is the microprocessor in low power mode, tx is the communication device in transmit mode, r is the communication device in receive mode and c_i represents other components, such as sensors and LEDs. The values of I_m, I_l, I_{tx} and I_r are device dependent. Throughout this thesis, Equation 7.2 is used giving the total energy E in mJ, the current in mA and $32768Hz$ is the default value for the on-chip timer for one second runtime on a $3V$ sensor. The values in the equation are provided in the TelosB datasheet [110].

$$E = (1.8t_m + 0.0545t_l + 19.5t_{tx} + 21.8t_r) \times \frac{3}{32768} \tag{7.2}$$

In [61] the authors developed a generic method to predict a node's energy consumption by capturing the interference patterns. The interference patterns of a specific deployment site are captured to enable the estimation of the node's energy consumption when the nodes are deployed at the same location in the future.

These energy consumption estimation solutions can be used to improve the network by using the information to reconstruct the topology.

7.4 MCRP Energy Estimation Implementation

ContikiMAC [33] radio duty cycling uses a transmission phase-lock optimisation to significantly reduce the length of a packet transmission. In the beginning, the sender sends the same packet repeatedly to the neighbour until it receives a link layer acknowledgement. The link layer acknowledgement is used as the indicator of the receiver's wake up phase. In the next transmission, the number of the transmissions will be shorter as the sender will send the packet just before the neighbour is expected to be awake based on the neighbour's wake up phase knowledge that it acquired previously. This ensures transmission efficiency, which as a result, reduces the network's energy consumption thus less radio congestion.

Powertrace is used to compute the energy consumption estimation of the network. However, the nodes do not have enough capability to compute their individual energy consumption. In order to estimate the energy taken from the sender to the receiver, each node sends their *energest* values to the LPBR regularly as MCRP has a centralised controller. This enables the LPBR to predict the energy drain if the routes have high interference or packet losses. The LPBR is able to compute the end-to-end energy consumption on each route and estimates the nodes' battery level based on the *energest* values. Each node sends the *energest* value of its packet transmission, packet forwarding and total time value that the radio has been on from the beginning to the LPBR for energy consumption computation. By doing that, the nodes knowledge of its energy level is kept at a minimum.

In order to calculate accurate energy consumption for a specific packet transmission, the unicast packet type is separated into normal unicast and control messages unicast. The unicast packet from the application layer (normal unicast) is set as *unicastMsg = 1*, which the value is 0 by default to represent other unicast packets. This allows the energy of the transmission packet to be calculated separately without including other control messages that could be sent right after or before the normal packet transmission. This is done to avoid inaccurate energy spent as control messages are only being sent periodically unlike the normal packets that are sent frequently. It also enables retransmit packets to be included as the current transmission packet's energy. This will alert the LPBR on the current condition of the node with much higher energy consumption than the usual energy per packet because of the retransmissions. The *unicastMsg* value is reset when the link layer acknowledgement is received or the maximum number of retransmission is reached.

Algorithm 2 Pseudo-code for packet's energy consumption

Notations
R is a node that is a Route
txE is the transmission period
$fwdE$ is the forwarding period
$totalE$ is the total run time
Pseudo-code
Generate the end-to-end routes taken by the node
if next hop node $= R$ **then**
  R is node's next hop
  check *R* next hop
  if $R = LPBR$ **then**
    all end-to-end routes found
    access energy table node txE, $R\ fwdE$
    compute energy consumption using Equation 7.2
  end if
else
  check *R* next hop *R*
  update the routes
end if

Algorithm 2 shows the pseudo-code of the implemented energy consumption calculation processes. The end-to-end routes are checked each time before the energy consumption is calculated. This is done to ensure the routes have not changed, otherwise, the routes are updated. The sender node checks the next hop node. If it is not the LPBR, which is the final destination, the forwarding energy, $fwdE$ is calculated as part of the packet's energy consumption. The next hops are included until the packet reaches the LPBR. The packet's

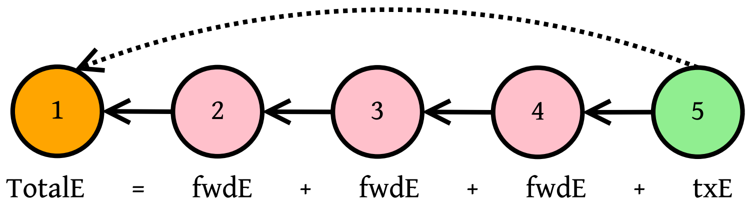


Figure 7.2: Transmission energy consumption

energy consumption $totalE$ can then be computed, which includes the transmission energy txE and the forwarding energy $fwdE$ consumed by the intermediate nodes.

Figure 7.2 shows an example of the *energest* values that are sent from the transmitting node (node 5) to the LPBR, which are the transmission txE_n, the forwarding $fwdE_n$ and the total time $totalE_n$ since it first booted to give the information of the total energy that has been used. The LPBR receives and keeps the values to calculate the energy consumption in a temporary table. As the network topology might change over time, the LPBR has to check the end-to-end routes before it can calculate the end-to-end energy consumption for a packet transmission. MCRP does not hardcode the routes because of this reason. However, the LPBR keeps the information of each node's next hop route, which is updated when there are changes. The LPBR has the knowledge of the whole topology.

In the example shown in Figure 7.2, the total transmission energy consumption for node 5 packet is the sum of txE_5 and the $\sum_{i=2}^{4} fwdE_i$ of all the hops before reaching the LPBR. The $fwdE_n$ is the node's transmission energy consumption that is used when it only forwards the packet. It is kept separately from txE_n to avoid confusion when the node n is transmitting its own packet after forwarding the other node's packet as the *energest* values might vary. Equation 7.2 is used to calculate the energy consumption in mJ.

7.5 Evaluation

This section describes the evaluation of the energy consumption as the result of the proposed multichannel protocol. The performance of MCRP is evaluated using the end-to-end energy, the energy consumption over time and the forwarding packets' energy.

7.5.1 Experimental Setup

Using the simulation layout as shown in Figure 5.1, the energy consumption in terms of the transmission per packet, the forwarding packet and the total energy used are computed to prove that multichannel protocol helps to prolong the network lifetime by using the energy more efficiently than in a single channel protocol network. Each node sends one packet per minute, 350 packets in total throughout the simulation period. Equation 7.2 is used to calculate the nodes' energy consumption. The energy consumption of each node is computed by the LPBR based on the information contained in the transmitted packet. Node 2, 5 and 15 energy usages are selected for comparisons as other nodes show similar result. Node 2 is one hop to the LPBR while node 5 is 2 hops and node 15 is 3 hops away. The maximum number of hops in the simulation is 3 hops. The result of a single channel protocol with no interference is used as the base case as it is the ideal energy consumption value. The results are also compared to the energy of a single channel protocol with moderate and extreme interference, and MCRP for multi channels protocol with mild, moderate and extreme interference.

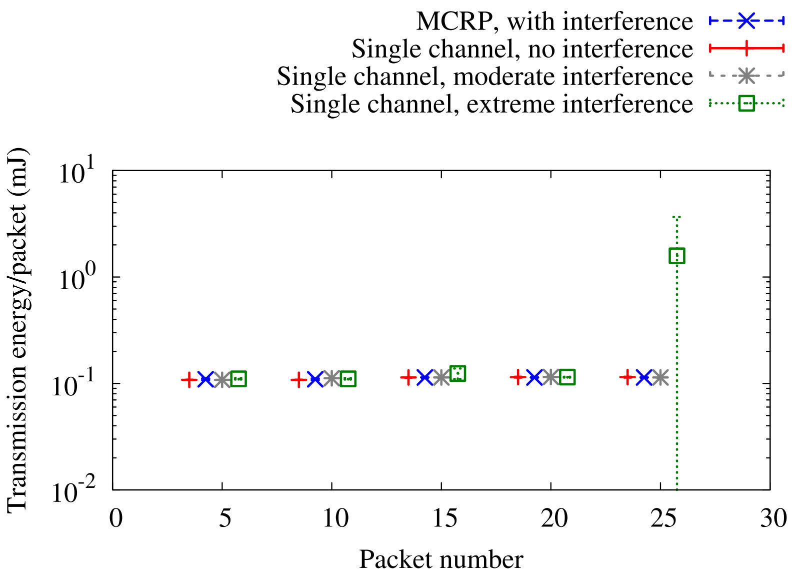
7.5.2 Energy Per Packet Performance

Figure 7.3 shows the transmission energy per packet for node 2, 5 and 15. From the figure, it can be concluded that less transmission energy is used when there is less number of hops. However, in a large scale network, the number of hops cannot be reduced as not all nodes would be in the range or directly connected to the destination node. Thus, the node's next hop should be selected carefully to avoid nodes that have higher interference rate.

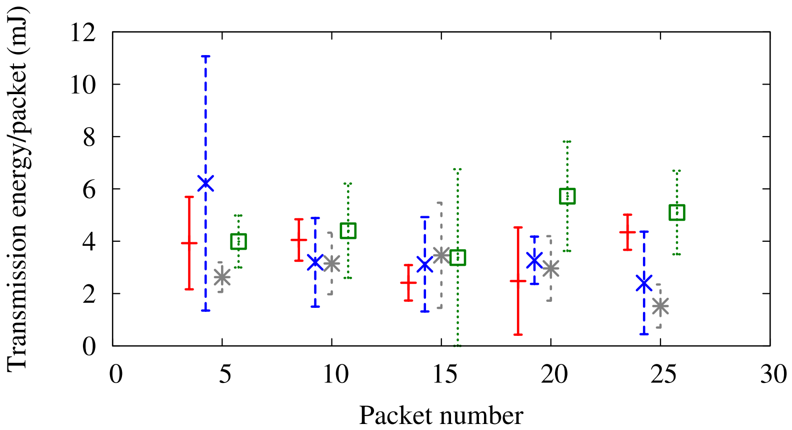
Node 2 energy consumption in Figure 7.3(a) for the 5th, 10th, 15th, 20th and 25th packet consumed approximately similar energy in all cases. As node 2 is one hop to the destination (LPBR), it was not affected by the interference except for a slight variation in the single channel protocol with extreme interference case for packet 25. Node 3 gives similar result to node 2 as it is also one hop to the LPBR.

Figure 7.3(b) and Figure 7.3(c) show higher values of energy that a packet requires from the sender (node 5 and 15) to the LPBR through 2 and 3 hops. This is because of the interference near to the nodes. The nodes are unable to detect the exact wake-up time for the nodes thus, the nodes have to transmit in a longer period to ensure the packet gets transmitted. In the one hop graph, the energy can be kept at minimum because the LPBR is always awake to accept packet as it is fully powered unlike the other nodes that have to switch the radio off when there are no transmissions and receptions taking place to save the energy.

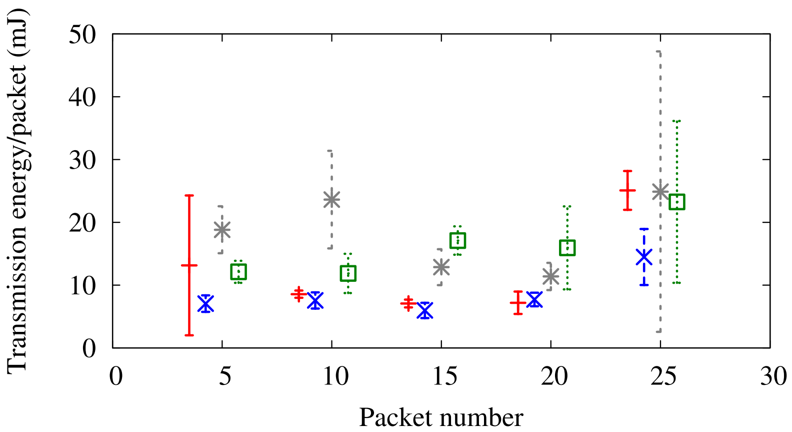
In both graphs, MCRP shows approximately similar transmission energy consumption to the base case. The transmission energy for a single channel protocol with moderate and



(a) Node 2 energy consumption



(b) Node 5 energy consumption



(c) Node 15 energy consumption

Figure 7.3: Simulation: Comparison of energy consumption per packet for node 2, 5 and 15

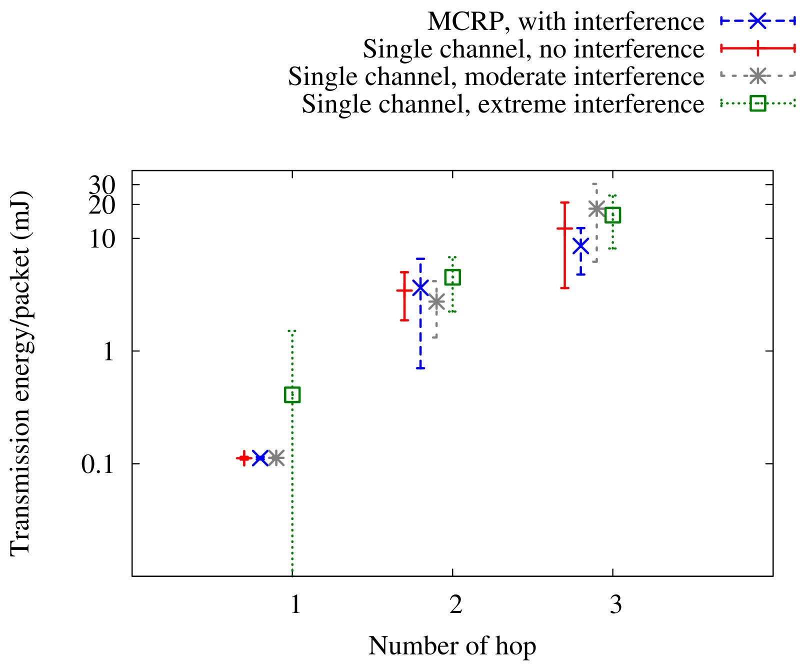


Figure 7.4: Simulation: The energy consumption per packet in different number of hops

extreme interference is slightly higher compared to MCRP in 2 hops. In 3 hops, the energy per packet in the single channel protocol with moderate and extreme interference are much higher than the energy used by the base case and MCRP. This shows that the energy per packet depends on the number of hops and the interference that affect the routes. Multi-channel protocol helps to mitigate the effect of interference, thus reducing the transmission energy taken to send a packet.

Figure 7.4 shows the transmission energy per packet for nodes that are 1, 2 and 3 hops away from the LPBR. It can be seen that less transmission energy is used when there is less number of hops.

In the 1 hop case, it can be seen that the nodes consumed approximately similar energy in all cases. As the nodes are one hop to the destination (LPBR), it was not affected by the interference except for a slight variation in the single channel protocol with extreme interference case. Nodes that are 2 and 3 hops away show higher values of per packet energy transmission. This is because of the interference near to the nodes. The nodes are unable to detect the exact wake-up time for the nodes thus, the nodes have to transmit in a longer period to ensure the packet gets transmitted. In the one hop graph, the energy can be kept at minimum because the LPBR is always awake to accept packet as it is fully powered unlike the other nodes that have to switch the radio off when there are no transmissions and receptions taking place to save the energy.

In the 1 and 2 hops, MCRP shows approximately similar transmission energy consumption to the base case. In 3 hops, the energy per packet in the single channel protocol with moderate and extreme interference are much higher than the energy used by the base case and MCRP. MCRP helps to reduce the transmission energy as the effect of using channels that have less interference.

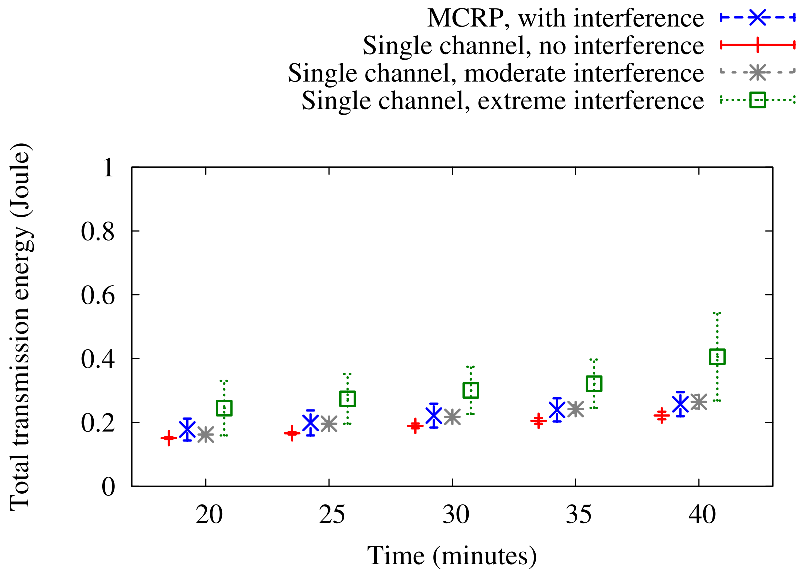
7.5.3 Energy Over Time Performance

Figure 7.5 shows the graphs of the three nodes' total energy consumption that the nodes took to send 25 packets (approximately 40 minutes) including retransmissions and control packets' energy. Figure 7.5(a) shows node 2 energy consumption, where it can be seen that in all cases, the total energy taken is approximately similar with a small increase over time. The single channel protocol with extreme interference case however, requires more energy consumption than in other cases. Figure 7.5(b) shows higher increase in energy usage over time in all cases. The reason for this is because node 5 has 2 other nodes that are using it as a forwarder. Node 5 (2 hops) uses higher energy when forwarding packet to the LPBR compared to node 2 (1 hop). Figure 7.5(c) shows lower energy consumption for node 15 compared to node 5 because node 15 does not act as a forwarder.

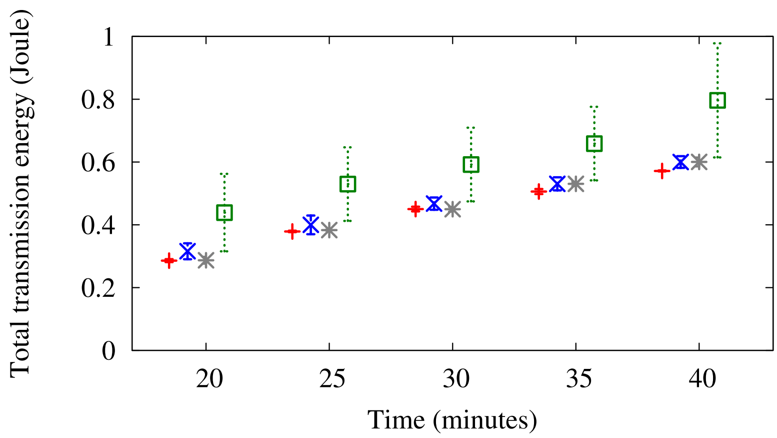
Figure 7.6 shows the total energy consumption for all nodes in the simulation. Node 2 and node 3 are one hop to the LPBR, nodes 4-7 are 2 hops, and other nodes are 3 hops away. For most nodes, the energy consumption is slightly improved when using MCRP than a single channel protocol with interference. This improvement can be clearly seen in the 2 hops nodes as these nodes use more energy during interference for retransmissions. If the retransmissions fail, the packet is dropped and the energy used during the retransmissions is wasted. The total energy consumption graph shows all energy from the packet transmission including failed packet's energy.

7.5.4 Forwarding Energy Analysis

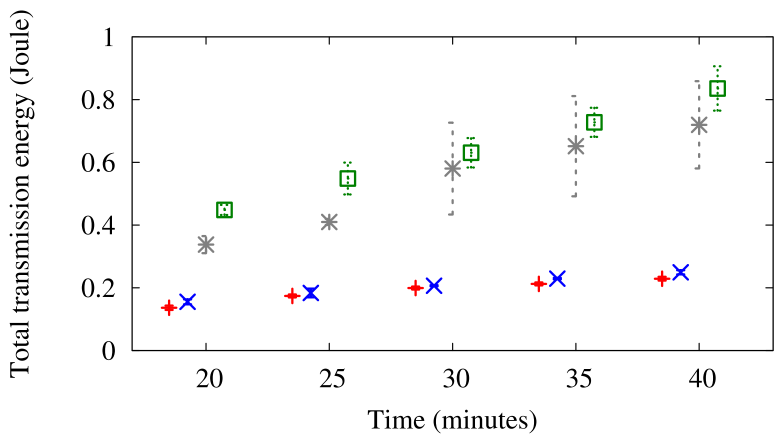
Figure 7.7 shows the energy used in forwarding packets for nodes 2-7. The other nodes in the simulation do not forward packets. Node 2 and 3 use less energy than nodes 4-7 as the nodes only need to check if the channel is being use by the other node before it can forward to the LPBR. The LPBR waits for the incoming packet thus the nodes could send the packet with less waiting time as the LPBR radio is always on. Based on the simulation layout in Figure 5.1, node 4 and 5 forward packets to node 2 while node 6 and 7 to node 3 as



(a) Node 2 energy consumption



(b) Node 5 energy consumption



(c) Node 15 energy consumption

Figure 7.5: Simulation: Comparison of total energy consumption for node 2, 5 and 15

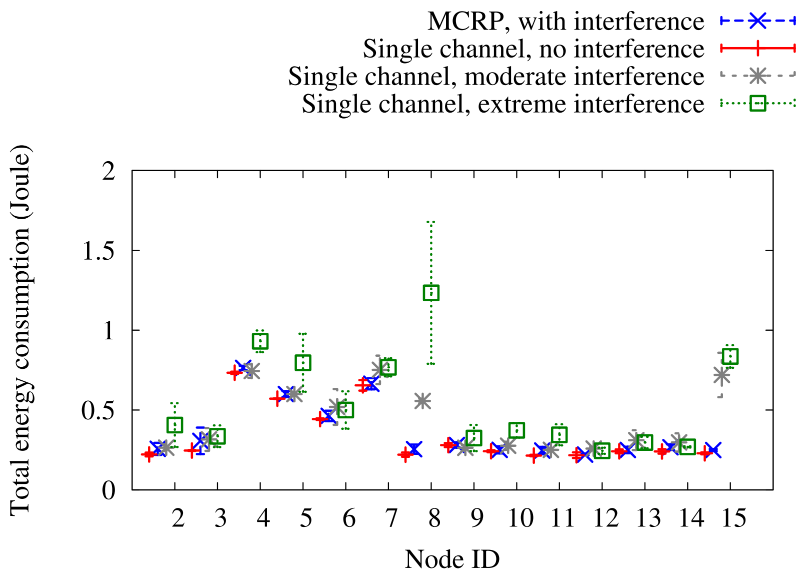


Figure 7.6: Simulation: The total energy of the nodes

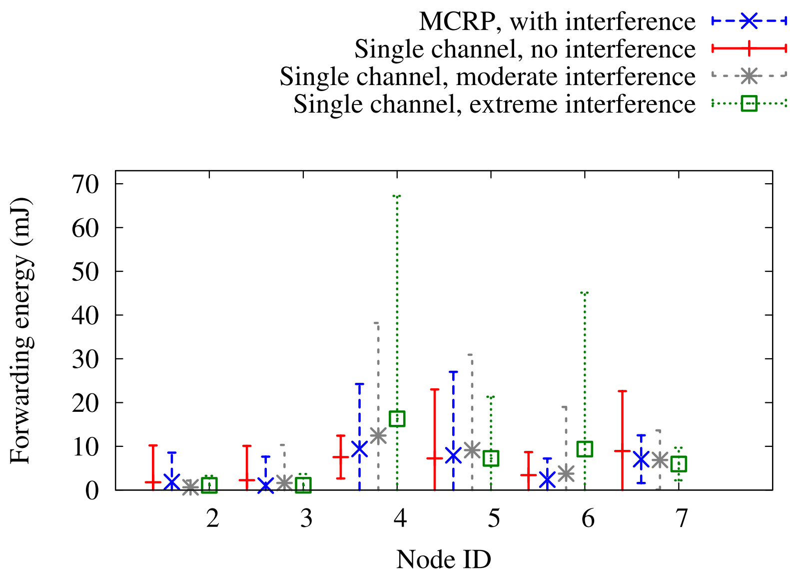


Figure 7.7: Simulation: The forwarding energy of the nodes

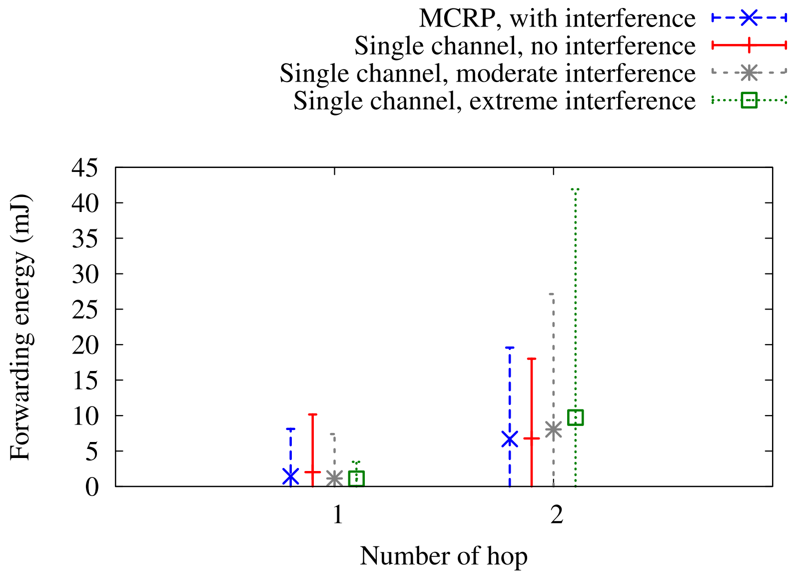


Figure 7.8: Simulation: The forwarding energy of one and two hops nodes

their next hop. Node 4-7 use higher energy than node 2 and 3 in forwarding packets as the nodes have more packets (from the children) to be forwarded. In order to be able to forward the packets, the nodes have to be awake for longer time and ensure the next hop is also awake and ready to accept the packets. Thus forwarding takes more energy consumption than an end-to-end packet transmission. By increasing the number of nodes, thus children, the nodes will use more energy in order to forward the packets. The forwarding energy consumption contributes to the most energy used by the nodes. MCRP helps to reduce the energy consumption, which can be seen in node 4 and 6 results than in a single channel protocol. In the base case, the energy consumption varies as the nodes are interfering with each other even without external interference during transmissions.

Figure 7.8 shows the energy used in forwarding packets for the 1 and 2 hops nodes. The 3 hops nodes in the simulation do not forward packets. The 1 hop nodes use less energy than the 2 hops nodes as the nodes only need to check if the channel is being use by the other node before it can forward to the LPBR. The LPBR waits for the incoming packet, thus the nodes could send the packet with less waiting time as the LPBR radio is always on. In order to be able to forward the packets, the nodes have to be awake for longer time and ensure the intermediate node is also awake and ready to accept the packets. Thus forwarding could require more energy consumption than an end-to-end packet transmission. By increasing the number of nodes, thus children, the nodes will use more energy in order to forward the packets. The forwarding energy consumption contributes to the most energy used by the nodes. Even though MCRP does not show a lot of improvement, it shows lower deviation compared to the single channel protocol with extreme interference. In the base case, the energy consumption varies, as the nodes are interfering with each other even without external interference during transmissions.

7.6 Summary

This chapter presents the estimated energy consumption calculation. Contiki implemented Powertrace, which is a software-based energy estimation. MCRP uses Powertrace that tracks the duty cycle, and uses the values to measure the estimated energy consumption. The simulation results showed that MCRP consumes less energy than the other cases as the effect of multichannel protocol. In order to increase the energy efficiency thus the network lifetime, MCRP needs to reconstruct the topology based on the energy consumption, the residual energy of the nodes and the link conditions gradually to avoid breaking any current connectivity. The next chapter explains MCRP tree optimisation in detail.

Chapter 8

Lifetime Energy Spanning Trees

8.1 Introduction

Based on the results in Chapter 5, it is clear that a multichannel MAC layer protocol alleviates the effect of interference. In order to increase the energy saving, thus prolonging the network lifetime, the topology has to be reconstructed.

There are many definitions of lifetime, as explained in Chapter 2. Lifetime in this thesis refers to the first node to run out of energy in the network. This definition is used because the first node to run out of energy could be the main node that is connecting the other nodes to the LPBR. The network would become non-functional despite the other nodes still being alive, because there are no other paths available to reach the LPBR.

The work described in this chapter aims to extend the WSN lifetime by improving the routing tree, by switching the minimum lifetime nodes from their initial paths to other paths, which could increase the lifetime of the network. When no new path is found, the tree is then the most feasible. The nodes are assumed to be on the favourable channels. The nodes have different transmission and reception channels. The LPBR has full knowledge of the nodes. The tree is built based on the nodes' residual energy and the link quality between the nodes and neighbours. The proposed lifetime energy tree is described in detail and the simulation results show the network lifetime improvement.

This chapter also describes the existing solutions in reconstructing the network; many of the studies do not consider the use of a multichannel protocol. The link conditions and the nodes' residual energy play a big role in ensuring an energy-efficient network. Chapter 7 shows that the energy consumption during transmission is reduced when using a multichannel protocol compared to a single channel protocol, when there is interference. However, frequent MCRP channel switching would require a higher overhead. Thus, this chapter proposes an improvement to the lifetime energy trees. When there are changes in the link conditions or the nodes' residual energy, the current tree is no longer energy-efficient, thus reconstructing the tree would help to ensure the network to be functional for longer.

8.2 Improving The Lifetime Energy Spanning Trees

RPL uses ETX, which is the expected number of transmissions to reflect the link reliability and the expected latency on the channel. The ETX value is calculated by the node in selecting the next hop route. In order to find the improved tree, it is assumed that RPL has selected the best routes and MCRP further improved the selected paths by switching to better channels for the transmissions. However, the current best paths do not take into account the nodes' residual energy. This could drain the battery of certain nodes more quickly than other nodes. RPL can reconstruct the tree as the result of MCRP. The routes might have better reliability in the new channel than previously, which reduces the number of retransmissions, thus reducing the energy consumption.

In order to maximise the network lifetime, MCRP has to consider swapping the paths for the affected node, one path at a time. The lifetime energy tree swapping has to take into account the number of children and descendants to balance the energy consumption in the network based on the residual energy of the nodes. There are three possible solutions that are considered: (a) swap the parent of node i, (b) swap the children of node i, and (c) swap the descendants of node i that are not the children and at least one hop away from node i. However, swapping the parent of the minimum lifetime node does not improve the node's lifetime as the number of children and descendants remains the same. Therefore, only option (b) and (c) are further investigated.

$$l_i = \frac{e_i}{(d_i+1)t_{ip(i)} + \sum_{j\in c(i)}(d_j+1)t_{ji}} \tag{8.1}$$

Equation 8.1 shows the lifetime energy tree calculation where l_i is the node i's current lifetime represented as a percentage of energy unit. The node's remaining energy e_i in this thesis is represented as a percentage of the available voltage. It can be represented in Volts, thus, l_i would then be in volts unit. However, for simplicity, percentage is used. d_i is the number of descendants, t_{ji} represents the number of transmissions on average from node i to node j, $p(i)$ refers to the parent of i and $c(i)$ is node i set of children.

Algorithm 3 describes the swapping processes based on the nodes' lifetime calculated

Algorithm 3 Pseudo-code for lifetime energy spanning tree algorithm

Notations
l_i is the node's lifetime
c_i is the number of node i children
d_i is the number of node i descendants
Pseudo-code
Form tree based on MCRP
Update battery level for all nodes
Update all nodes l_i, c_i, d_i
minimum $\leftarrow$ 0
previousSwapNode $\leftarrow$ 0
while node $\neq$ previousSwapNode **do**
  Find node with minimum l_i
  List all potential c_i and d_i swap
  if c_i and d_i swap $l_i >$ minimum **then**
    Recalculate all nodes l_i
    if all new nodes $l_i >$ minimum **then**
      Update tree
      New tree is improved
    else
      Revert to previous tree
    end if
    previousSwapNode $\leftarrow$ node
  else
    Current tree is improved
  end if
end while

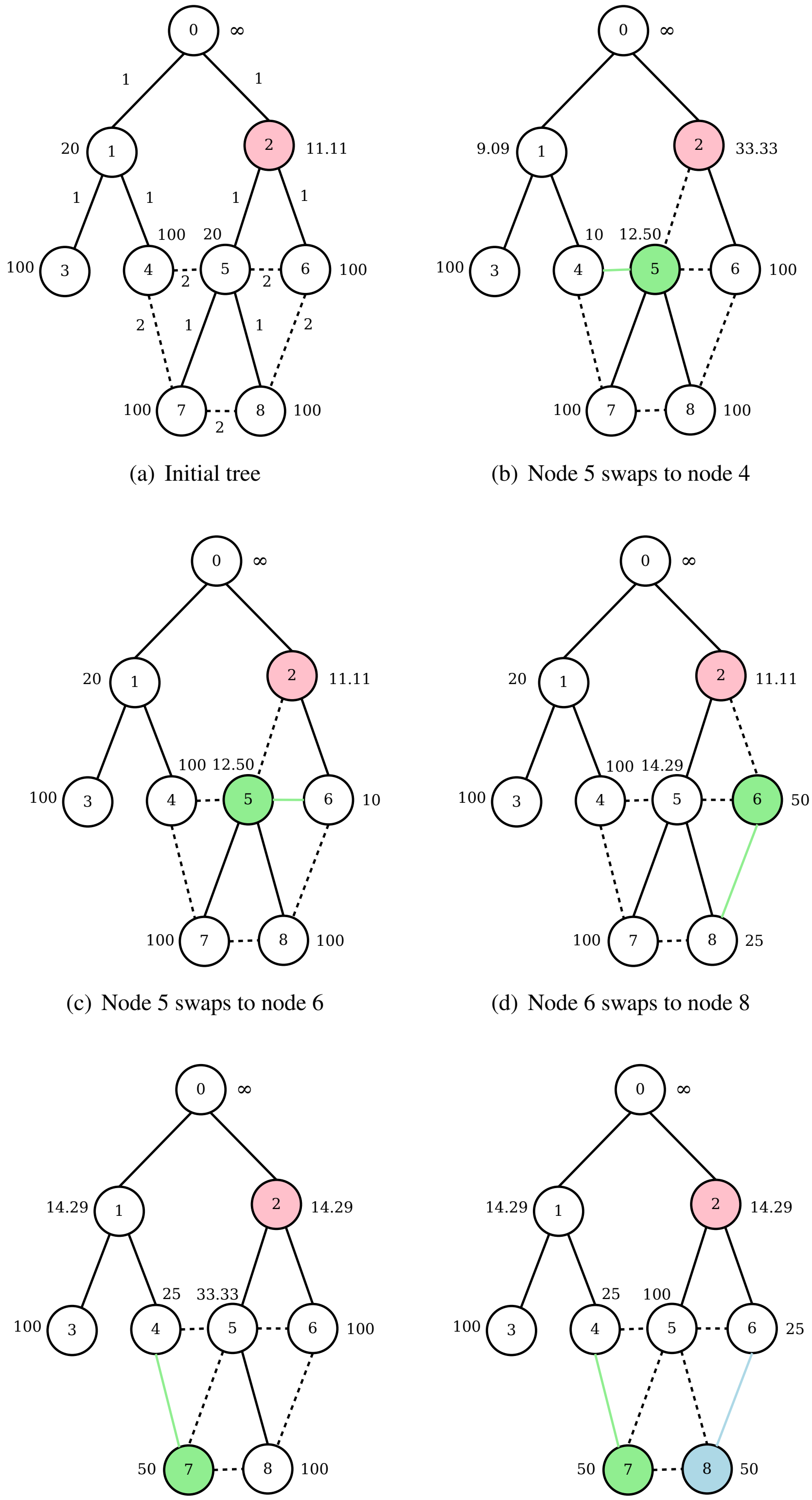
in Equation 8.1. It considers all available paths between the nodes and shows all potential topologies before deciding on the improved tree. It is assumed that all nodes' residual energy and the paths are known. Both the nodes' battery and the link conditions can deteriorate over time. However, it is assumed that the current selected paths are the favourable routes selected by MCRP, thus, only the battery level of the nodes is the variable.

The algorithm calculates the nodes' lifetime, l_i, based on the battery level and selects the node with the minimum lifetime value. Each of the node's children, c_i, potential paths to other nodes, is first considered. The network lifetime is calculated with each path change. If any of the nodes' new lifetime calculated has a value below the initial minimum lifetime value, the swap is cancelled. If the tree shows improvement, the new tree is selected. Regardless of the result, it then tries with the next child of the node, c_{i+1}, to see if the lifetime could be improved. If the node's children have children (descendants of the node, d_i), the descendants are required to run the swapping one after another after all children have tried swapping. Once the node has considered all its children and descendants, the algorithm searches for the next node that has the minimum lifetime value. If the node selected is the same as the previous node, the swapping stops as it could lead to endless cycle of swaps. The tree is then selected as the improved lifetime tree. The network is considered as balanced in terms of the lifetime, which means, the number of nodes and descendants connected might not be fairly distributed as the battery level vary in each node.

Figure 8.1 is an illustrative example to explain the algorithm proposed. Assume that the tree formed in Figure 8.1(a) is the current tree after running MCRP processes. Each node is labelled with the current lifetime, represented in percentage for simplicity. Each node is assumed to have full battery level e_i, represented as 100%. It can also be represented in volts or joules. The lines between the nodes represent routes in different channels where dotted lines are the potential routes and the solid lines are the current routes. The values represent the link conditions in terms of the number of successful expected transmission between the two nodes. As an example, when the value is 1, it means that it requires only one transmission for the packet to be successfully received while if it is 2 or more, it has to be retransmitted once (or more) depending on the value. The values of the links are the expected transmission taken only for the upwards route as the links downwards could have different values due to the different transmission and reception channels on each node, thus different link quality. The transmission and reception channels of a node cannot be the same to avoid interference with nearby nodes.

Figure 8.1(a) shows that node 2 has the most descendants, which consequently reduces the node's lifetime as it has to forward more packets than any other node. Initially, the topology is formed based on the least computed transmissions value on the paths. In order to optimise the tree, the overall network lifetime is considered, where paths that are not the least value could be chosen as the route as it prolongs the overall functionality of the network. In this example, node 2 has the minimum lifetime. It can be increased through swaps.

There are several potential swaps to improve node 2 lifetime that includes both the children, which are node 5 and 6, and the children of children, node 7 and 8. Figure 8.1(b) shows node 5 swaps to node 4 instead of its initial node 2 and the network lifetime is calculated using Equation 8.1. Node 2 lifetime improves from 11.11% to 33.33%, however, node 1 has a lifetime of 9.09%, which is lower than the previous node 2 minimum value of



(a) Initial tree (b) Node 5 swaps to node 4

(c) Node 5 swaps to node 6 (d) Node 6 swaps to node 8

(e) Node 7 swaps to node 4 (f) Node 8 swaps to node 6

Figure 8.1: Graph of the bidirectional paths in a WSN

11.11% as the result of swapping. In this swap, node 1 has 5 descendants while node 2 only has one when it initially had 4, which is the reason for node 1 lifetime to decrease by more than half of its previous lifetime value. The network reverts to the previous topology that has better overall lifetime than the new swap. The algorithm tries to improve the minimum node lifetime without deteriorating the other nodes to below minimum. This allows the network lifetime to be consumed at similar rate.

Node 5 then tries and swaps to node 6 as shown in Figure 8.1(c). However, node 2 shows no improvement and node 6 lifetime decreases to 10%. The network reverts to the previous tree. As there are no other paths available to node 5, node 6 which is node 2 child is selected for swapping. Figure 8.1(d) shows node 6 swapped the path to node 8. However, the potential topology is not improved as the minimum lifetime is still 11.11%. Node 2 then considers swapping its descendants node 7 and 8 as there are no children left.

When node 7 is swaps to node 4 instead of node 5, the tree is improved. It can be seen in Figure 8.1(e) that the tree is more balanced and node 2 lifetime is prolonged. As the result of swapping, node 4 lifetime is reduced as the path from node 7 to node 4 is not the smallest path value. The tree is updated as the current improved tree. It is not yet the final improved tree because node 8, which is another node 2 descendant has not been checked. If node 8 swaps does not improve the tree, the swap from node 7 is chosen as the final improved tree. Another potential swap is shown in Figure 8.1(f) where node 8 is connected to node 6 instead of node 5. In both cases, node 2 lifetime is increased and all nodes' lifetime are above the minimum value. The tree in Figure 8.1(e) is selected as the final improved lifetime energy tree in maximising node 2 lifetime. Further investigations are required in order to decide the criteria on an improved tree when there are several good topologies to be selected.

Node 1 is then selected as the minimum lifetime as node 2 cannot be selected again to avoid unnecessary repetition. Improved tree from the potential swaps for node 1 is not found thus the tree is said to be improved. In the algorithm, the same node cannot be swapped again right after its previous swap. This is done to avoid oscillation, which would produce similar result. The node however, could swaps in the next round as the other node swap would have changed the topology.

The swaps are assumed to happen once until the network stops functioning, thus the overheads are negligible. The swapping calculations and decisions are made by the LPBR due to sensor nodes' limitations and constraints. The LPBR informs the specific nodes of the final swapping if it needs to take place. In terms of the energy cost, the cost is negligible as the swaps are infrequent and being controlled by the LPBR.

8.3 Evaluation

This section describes the evaluation of the following performance metrics: (a) the average number of switches to form the improved tree and (b) the impact of swapping on the network lifetime.

8.3.1 Experimental Setup

The lifetime energy tree algorithm is simulated in C programming language. The number of sensor nodes considered is between 10 to 500 and each node is randomly assigned an initial energy between 50% to 100%. The link conditions are also randomly assigned the value between 1 to 10, where smaller number indicates better link condition as it requires smaller number of retransmissions. In this setup, the channels are fixed, assuming that the current channels that the nodes are listening and transmitting on, are the most favourable selected channels from the previous MCRP processes. RPL builds the initial tree based on the ETX value, which is then further improved by MCRP. In order to avoid all the nodes from directly connecting to the LPBR, each node could route to a minimum of $1/10$ and $1/30$ of the total number of nodes. This allows the node to have alternative parents (thus paths) for the swapping processes and hops to reach the LPBR. The node is not necessarily connected to all $1/10$ and $1/30$ nodes. The paths between the nodes are selected based on RPL and MCRP processes. These values are selected for swapping in the case where (a) the nodes are closely together ($1/10$), and (b) the nodes are spread out with minimum connections to the other nodes available ($1/30$).

8.3.2 Average Number of Switches

Figure 8.2 shows the standard deviation and average number of swaps using lifetime energy tree algorithm. The standard deviation on the x axis is slightly shifted to avoid overlapping. It can be observed that there are more swaps on average when there are more sensor nodes in the network in both connection cases. However, nodes that could route to $1/10$ of the total sensor nodes showed slightly higher number of swaps than in the $1/30$ connection. The reason for this is because in $1/10$ connection, it has more neighbours, hence many potential parents to select.

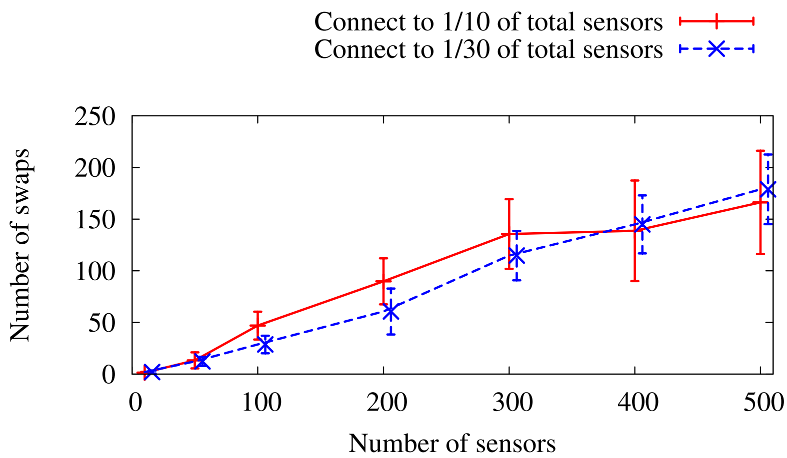


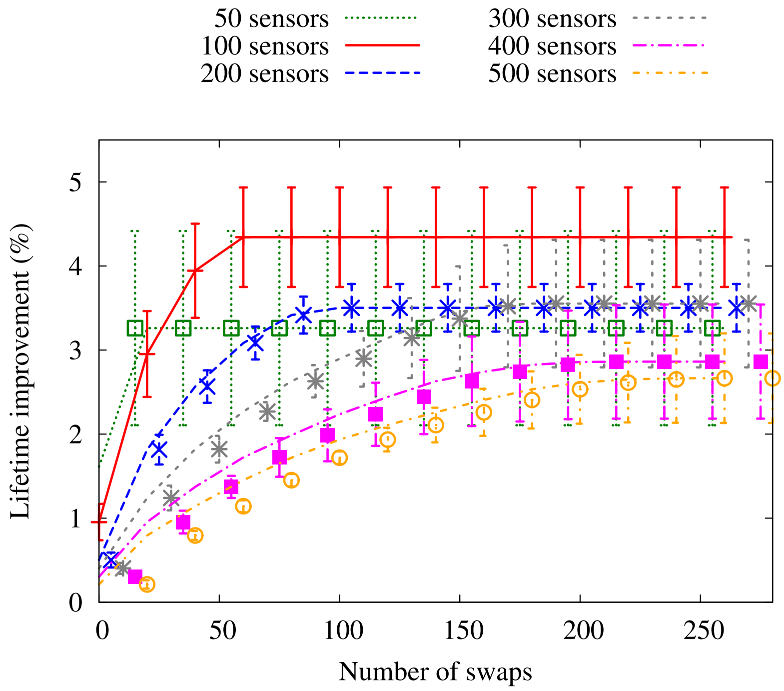
Figure 8.2: Simulation: Average number of swaps

In a smaller network, the sensor nodes are limited by the number of potential parents. This prevents the nodes from swapping as the potential parents might not have any improvement. This can be seen from the number of swaps in a 50 sensor nodes network where the average number of swaps is less than 10. This is because each node has 5 (in $1/10$) and 2 (in $1/30$) potential parents to select from unlike in the 500 sensor nodes network, which each has 50 and 17 potential parents. However, as there are more sensor nodes, it takes longer to find the best parents as the swaps will consider all nodes that are within the range.

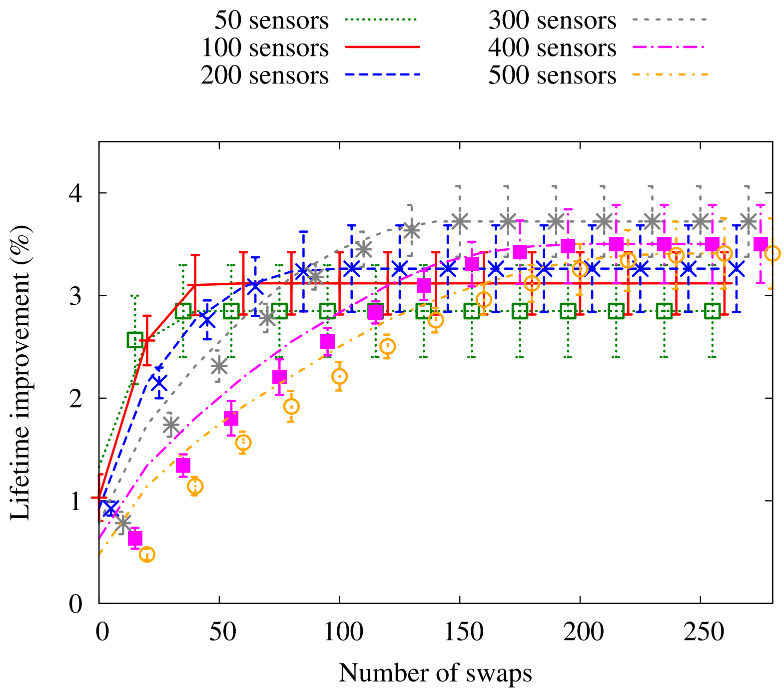
This experiment does not reflect the condition in the real-world where the sensor nodes could be scattered with more or less available range to the other nodes depending on the application, which means more nodes can directly be connected to the LPBR without hops in between. This however, represents a reasonable connection between the nodes to allow swapping and alternative nodes if the current forwarding nodes' values are at a minimum for the experiment.

8.3.3 Impact On The Network Lifetime

Figure 8.3 shows the improvement in the node thus network lifetime by maximising the minimum and the number of swaps required in 10 runs in 6 networks with 50, 100, 200, 300, 400 and 500 sensor nodes with different degree of connection. The standard deviation on the x axis is slightly shifted to prevent the error bars from overlapping. It is observed that the node's lifetime decreases with an increase in the number of sensor nodes in the network. The reason for this is in a larger network, there are a higher number of descendants and each connection has its own path values. By taking into account these variables, the number of nodes affect the whole network lifetime. While a higher number of nodes allow



(a) Connect to 1/10 of total sensors



(b) Connect to 1/30 of total sensors

Figure 8.3: Simulation: Comparison of the number of swaps and lifetime in different network scale

more alternative routes, it also consumes more energy as there are more connected nodes. Smaller network however, has limited number of possible swaps, which does not improve the network lifetime.

Figure 8.3(a) shows the number of swaps and the lifetime when the nodes are connected to $1/10$ of the total nodes in the network. In the 50 nodes network, the number of swaps is less than 10 before it reaches the maximum lifetime for the whole network. As the number of sensor nodes in the network increases, it takes more swaps before the tree is improved. In the 200 nodes network, the number of swaps is around 100 swaps and the lifetime is improved from the minimum of 0.5% to 3.1%. In 500 nodes network, it takes more swaps, around 200 swaps for the lifetime to be maximised from 0.3% to 2.5%.

In the 50 and 100 sensor nodes networks, they showed different results than expected with the other types of networks. 50 nodes networks lifetime showed improvement but have less lifetime improvement than the other larger number of sensor nodes networks. The reason for this is because of the lack of paths to be selected, which could increase the overall lifetime. 100 sensor nodes networks have more selection to swap, thus higher lifetime value. However, as the number of nodes increase, each node might have more children and descendants, which reduce the overall lifetime.

Figure 8.3(b) shows similar improvement in the $1/30$ connection case. However, it can be seen that the maximum lifetime values in the figure for 100 sensor nodes network is slightly less than in Figure 8.3(a). This is because the networks have lesser potential parents and paths to select from. The tree is limited by the number of connections. The other large networks have similar maximum lifetime values in both figures.

Figure 8.4 shows the comparison between the initial and improved tree in both cases. The standard deviation on the x axis is slightly shifted to the left and right to prevent the error bars from overlapping. MCRP swapping prolongs the network lifetime, which shows an increase from the initial lifetime. Smaller networks have high initial lifetime values compared to larger networks. However, larger networks have better lifetime improvement than the slight improvement in smaller networks.

In the initial trees, it can be seen that when there are more sensor nodes in the network, the lifetime values are decreasing. This shows the importance of finding the improved tree as the results showed that the lifetime can be improved by approximately 3% when initially, in all networks, the minimum sensor nodes have less than 1% lifetime. The increase enables

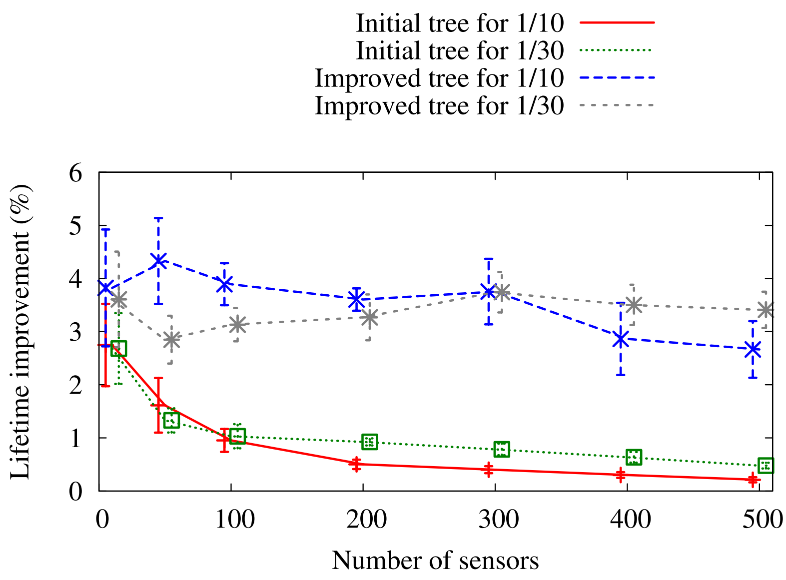


Figure 8.4: Simulation: Lifetime of the improved energy tree

the network to remain functional slightly longer than initially.

8.4 Discussion

The proposed lifetime energy spanning tree is different than the other existing tree in terms of the use of multichannel protocol as part of the routing tree decision. The previous chapters showed that multichannel protocol helped to increase the communication rate. By considering multichannel protocol and the node's energy in routing decision, better topology can be formed. In MCRP, the network is communicating on better channels. As MCRP does not consider the node's energy as part of the process, node that has better path could be overloaded for transmissions, which resulted in quicker energy drain.

The lifetime energy spanning tree uses three critical information of the nodes and paths that influence the network lifetime to find better and improved topology. The information is the node's energy, path condition and the number of reachable nodes and descendants. By reconstructing the network topology based on the information, better topology can be found, which allows all the nodes to be functional longer than initially.

The lifetime energy spanning tree is done at the LPBR, which has the knowledge of the whole network and the computational ability. This means that the LPBR could attempt different options and combinations, such as connecting the node to a different parent node or descendants in order to prolong the network lifetime.

Even though the lifetime energy spanning tree showed the network lifetime to be improved by 8 times than the initial lifetime, the topology found is not the optimal result. This is because the proposed tree checks one node at a time. Checking several nodes at the same time might enable optimal tree to be found. However, the exhaustive search would be computationally infeasible for large networks. It is NP-hard optimisation problem. It is hard to approximate and blindly selecting the nodes can result in bad tree, which might not improve the network lifetime.

As the definition of lifetime used in this thesis is the first node to run out of energy, the proposed lifetime energy spanning tree concentrates on maximising the minimum node energy. The definition is chosen to depict the worst case scenario, where the minimum energy (nearly non-functional) node is the main and only node connected to the LPBR. If this node fails, the whole network is affected as no transmissions could get to the LPBR. However, depending on the application and the density of the nodes in the area, it might be acceptable to sacrifice some sensor nodes, such as setting a threshold of the minimum number of available nodes required to maintain the network connectivity and functionality.

Further directions to improve the lifetime topology are discussed in the Future Research Directions section in Chapter 9.

8.5 Summary

This section presents the lifetime energy tree in addition to multichannel protocol. The lifetime energy tree algorithm is implemented in C. It takes into account the node's energy, path conditions and the number of reachable nodes and descendants to estimate the lifetime value. The algorithm tries to balance the energy consumption, thus lifetime, across all nodes by swapping nodes to different potential parents that could results in higher lifetime value. The sensor nodes could then consume the energy at similar pace. The simulation results showed that by maximising the minimum lifetime, the network is able to remain functional longer than it was initially until the first node fails.

8.6 Existing Energy Efficient Tree Solutions

There are many studies that were looking into improving RPL by including the energy as the metric in selecting a next hop neighbour [59, 21, 55, 113, 10, 18, 98] as RPL is designed as a low complexity routing protocol that minimise the sensor nodes' memory requirements and reduce the overheads by using the trickle timer to reduce the number of control packets over time. There are also studies that instead of concentrating on the energy directly, increases the network lifetime by distributing the communication load in the network, such as in [76, 28] rather than overusing certain nodes that are either closer to the sink or selected as the best route to get to the sink.

RPL is a routing protocol that builds the topology based on the Objective Function (OF) that specifies the routing metrics and constraints for path calculation, which are translated into rank. The rank value is used to select the next hop in order to optimise the routes. The OF is application dependent as RPL does not define any specific OF. This separation allows new metrics and constraints to be defined to fulfil the specific application and network's optimisation criteria. A routing metric is used to evaluate the path cost. The routing metrics can be categorised into link and node metrics [117]. In node metrics, it can be the node's state, which provides information about the node's characteristics, energy such as selecting nodes with higher residual energy or hop count. In link metrics, it includes the link throughput, latency or link reliability such as ETX. RPL provides the list the metrics that could be used. However, the implementation is left to the application.

Typically, ETX, which is the expected number of transmissions until a link-layer acknowledgement is received is used other than hop count. It influences the link reliability and latency. The minimum value of ETX is selected at each hop, which can indirectly be translated as the minimum energy path. However, if all packets are being sent on the minimum energy path route, the nodes along the path are likely to have higher energy drained thus reduced lifetime than the other nearby nodes. Thus, it is important to have energy balance nodes to ensure that the nodes consume the same quantity of energy in order to increase the overall network lifetime.

Existing studies on energy based RPL and load balancing RPL are explained and summarised in the section below. In all studies, at least two metrics are being considered in order to balance the network in terms of improving the nodes' energy consumption while maintaining a high value of packet throughput and ensuring energy and load balanced network.

Compared to the proposed lifetime energy spanning tree solution, these studies concentrate mainly on improving the routing tree in terms of the energy without considering the impact that multichannel could have on the protocol. MCRP lifetime energy spanning tree took into account the improvement due to the multichannel protocol, which shows improved communication rate as the number of retransmissions and packet loss are reduced as

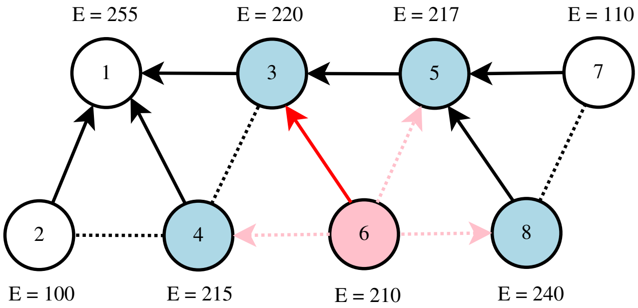


Figure 8.5: Energy based path cost

the communication is on better channels with less interference. It then further reconstructs the topology of the multichannel nodes based on the node's energy, path conditions and the number of reachable nodes and descendants to prolong the network lifetime.

8.6.1 Energy-based RPL

In energy-based RPL studies, the energy or battery level is the used as one of the performance metrics. However, the energy or battery model varies in different studies because of the complexity in acquiring the battery level in real-time.

Kamgueu et al. [59] designed an OF for RPL that uses the node's remaining energy as the metric during the parent selection of the topology. It aims to select nodes with higher remaining power level as the path for transmissions. The implementation uses a battery theoretical model [81] to estimate the node's battery lifetime at runtime. The OF concentrates on the node's battery level estimation, path cost and node's rank computation in selecting a parent. The node that advertises the maximum greatest path cost is selected as the parent. The maximum path cost from the node to the sink is computed as the minimum node energy level. Figure 8.5 shows a simple example of the protocol where node 6 selects node 3 as the path as it has the higher path cost. Even though node 8 has the maximum path cost, it cannot be selected to avoid loops. RPL employs a mechanism to detect loop through *rank*, which is the distance from the node relative to the other nodes, with respect to the root is known. Node 8 is further away than the node, thus it cannot be selected as node 6 parent.

Chang et al. [21] improves the RPL routing protocol by combining the expected transmission count (ETX) and the remaining energy metrics in path selection. However, using the lowest energy consumption path would result in a bottleneck because of the unbalanced energy consumption due to unbalanced communication traffic load as these nodes may consume more energy than the other nodes as the nodes are actively used as the next hop. This

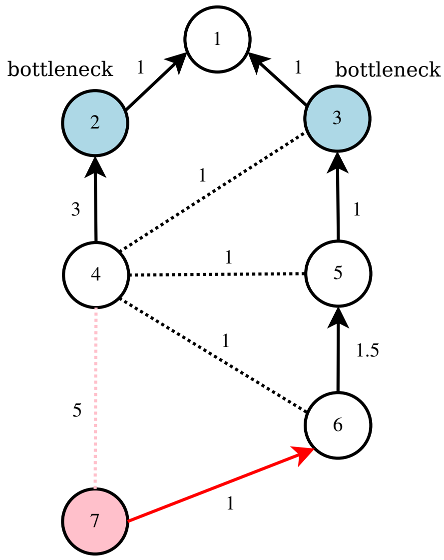


Figure 8.6: Expected Lifetime (ELT)

does not improve the network lifetime and decreases the network coverage as certain nodes that are overused would have shorter lifetime. Thus, the author introduced a switching mechanism in order to balance and optimised the paths and residual energy of the nodes. Each path is given a routing score in order to be selected as the next hop route, which depends on the ETX and the node's residual energy. The residual energy is calculated by deducting the energy consumption of transmission and reception from the battery.

Iova et al. [55] defined a new metric called Expected Lifetime (ELT) that estimates the lifetime of the nodes that had been identified to be the first ones to run out energy. ELT uses the nodes' residual energy, the link quality and the current traffic conditions in order to maximise the minimum nodes lifetime instead of minimising the energy consumption. ELT is based on ETX where it takes into account the link quality and tries to balance the traffic load by constructing paths of the same energy consumption in the function of nodes available energy on the path. The packets are routed in the way that the nodes that are most constrained with the least residual energy are avoided to maximise the nodes' lifetime. By doing so, the topology is energy balanced with all the nodes having similar level of residual energy to prolong the network lifetime. ELT showed similar result to ETX in terms of reliability and delay, and further improvement in terms of building an energy balanced topology, which reduced the nodes' energy consumption. Figure 8.6 shows an example where node 2 and node 3 are the bottleneck nodes. Node 7 has the option to send packets through node 2 or node 3. Node 3 has more descendants than node 2, however, node 2 has bad links. Node 7 selects the paths through node 3 as it maximises the minimum ELT between all nodes than through node 2.

Todolí-Ferrandis et al. [113] implemented a new OF that focused on optimising the energy in nodes by enabling the nodes to change their parents based on the neighbour nodes' residual energy. The remaining battery value is directly poll from the nodes in real-time, which helps to avoid the need to record and manage the nodes change of state values used to compute the energy consumption. The new OF aims to equalise the energy load in the network in order to ensure that the nodes' batteries level deplete equally fast in the network. The OF sets a threshold of 5% to avoid frequent changes, similar to the one used in ETX for a stable network. The nodes' batteries are checked each time before any changes take place.

Barbato et al. [10] proposed a new routing protocol that is energy aware and resource oriented based on RPL called Resource Oriented and Energy Efficient (ROEE). ROEE uses two metrics that are the energy consumption and the battery index. Energy consumption is selected as one of the metrics as it shows the amount of energy used by the node. This enables nodes with the highest residual energy to be selected as the routes, which resulted in an increase of the network lifetime. The battery index keeps track of the node's power consumption and vulnerability in each transmission, reception, idle and sleep states. It detects nodes that have drained energy. ROEE also uses the resource availability information in defining the rank for node selection in addition to energy consumption and battery index metrics. This allows the protocol to assign roles thus paths from the node to the root, which the node can reply to the specific request. ROEE uses its energy aware routing metrics to retrieve the requested resource to improve the network's energy efficiency.

Capone et al. [18] proposed a composite metric called Lifetime and Latency Aggregateable Metric (L^2AM) that considers the nodes' energy consumption and reliability through ETX in order to prolong the network lifetime by balancing the nodes' energy consumption. L^2AM combines multiple routing metrics into a composite in order to optimise the overall network and nodes' performances. Exponential Lifetime Cost (ELC) metric is proposed, which takes into account the link transmission power and the node's residual energy in deciding the routes. ELC is simplified and called Fully Simplified Exponential Lifetime Cost (FSELC), which keeps the same behaviour that ELC is intended. L^2AM carries

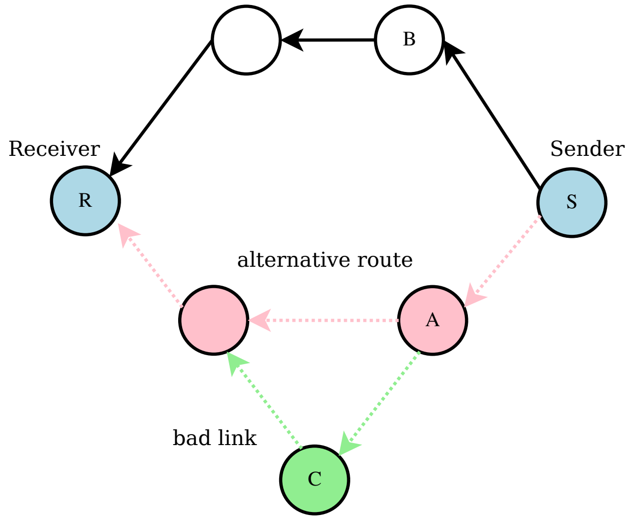


Figure 8.7: Neighbourhood metric

information about the link quality and latency through ETX, and the node's lifetime, which includes the cost to inject a link layer message into the communication layer represented by the FSELC metric. The parent or path is selected based on the L^2AM, computing the minimum cost paths. L^2AM composite metric value is advertise to the neighbours through DIO messages. The node will switch to alternative routes and reselect the preferred parent when the energy is depleted based on the L^2AM metric.

Sharkawy et al. [98] proposed a Context-Aware Objective Function (CAOF) that enables the parent selection to be based on the nodes' capabilities, resources availability which is the battery level and the location to the sink node. CAOF defined the battery model to represent the maximum number of seconds that the modelled battery can hold. The time taken during transmission and reception are subtracted from the battery value. CAOF computes the parent selection based on the battery level, node's duty cycle and the collocation with the sink. This allows a longer network lifetime as CAOF distribute the loads over different parents or routes based on the resource usage, which is the battery level.

8.6.2 Load balanced Routing

In load balanced routing, the workload is distributed in the network, which as a result, distributes the energy consumption across the nodes. These studies however, do not use the energy or battery level as the performance metric but instead use the nodes' workload value.

Delaney et al. [28] introduces *neighbourhood metrics*, which is used with the RPL ETX to create a new metric for routing selection that reflects the next hop nodes' conditions.

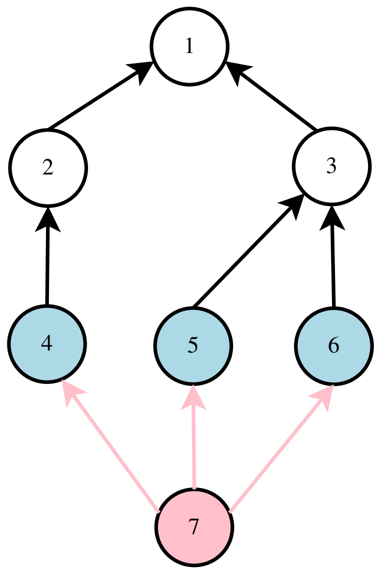


Figure 8.8: LB-RPL

It uses routing through good neighbourhood, which provides alternative routes instead of concentrating on a single good path to ensure that the workload are widely spread and no specific nodes are being used excessively. The neighbourhood metric uses the information regarding the quality of the surrounding neighbourhood, which are the forwarding path value and the neighbourhood influence on the node in making a decision. The next hop neighbour that is not selected as the parent becomes the alternative route if the current path is unavailable. The neighbourhood metrics allow a set of forwarding routes to be used to enable network load distribution, which as a result, helps to reduce the nodes' energy consumption and improve the network load balancing. Figure 8.7 shows an example of the metric where the sender transmits through node B to get the receiver. However, if the current path becomes unavailable, it switches to the alternative route, which is through node A. Node C is not selected as an alternative route as node A shows better path. However, if the route improves, it can be selected as an alternative.

Liu et al. [76] proposed a new protocol called Load Balanced Routing Protocol (LB-RPL), which is based on RPL. LB-RPL takes into account the workload distribution and the link layer communication qualities to achieve a balanced workload distribution in the network. LB-RPL adopts the RPL protocol tree routing procedure, which uses the control messages and incorporate the load balance mechanism to enable paths to be dynamically selected based on the workload distribution. LB-RPL delays the DIO control message transmission and starts a timer that is proportional to its workload in the previous period to signal workload imbalance. The DIO packet is transmitted when the timer expires. As a result, the node is less likely to be selected as the next hop for packet forwarding, thus the node's heavy workload is alleviated. Figure 8.8 shows an example of the protocol where node 7 has three potential next hop parents. LB-RPL selects the top two parent nodes, which as an example, node 4 and 5. The packet load from node 7 is distributed among the parents (node 4 and 5) according to the link quality between node 7 to each parent. Node 4 and 5 could transmit different number of load from node 7.

8.6.3 Comparison and Discussion

The energy-efficient routing protocols reviewed are summarised in Table 8.1. It can be concluded that the energy consumption and workload need to be balanced in the network in order to optimise the throughput while increasing the overall network lifetime. The important factors that influence the decisions in an energy-efficient routing are:

1. **Route metric** - All of the studies use several metrics in order to optimise both the residual energy and packet transmissions with most of the studies use ETX in addition to another metric, which usually is the residual energy level. Other metrics such as the location to the sink and resource oriented were also considered to increase the efficiency by specifying certain nodes instead of the whole network.

2. **Battery model** - The studies in [59, 21, 55, 113, 10, 18, 98] take into account the battery level by direct polling or using other alternatives such as subtracting the energy consumption from the battery level to estimate the residual energy. While direct poll enables the exact battery level to be known, it is not feasible in all conditions and locations. Subtracting the energy consumption from the known battery level, on the other hand, increases the complexity in computing the nodes' residual energy as the nodes have resources constraint. Better way in calculating the residual energy is required to get an accurate estimate of the battery level thus the network lifetime.

3. **Load balancing** - The studies in [55, 28, 76] take into account the packets transmission that is overloading the best path by helping to move the workload from overusing individual nodes, which as a result, balanced the energy on the nodes. Unbalanced workload distribution could lead to shorter network lifetime as the energy is depleted quicker for certain nodes. Load balancing effects the energy consumption of the

Protocol	Route Metric	Path Selection	Battery Model
Energy-based OF	Residual energy	Minimum energy level	Battery theoretical model [81]
Energy-oriented routing	ETX, residual energy	Switching mechanism	Deduct energy consumption from the battery
ELT	based on ETX; residual energy, link quality, traffic condition	Maximise minimum nodes	Multiply transmission power with transmission traffic
OF Energy	Residual energy	Remaining energy	Direct poll
ROEE	Energy consumption, battery index, resource availability	Highest residual energy	Deduct energy consumption from battery
L^2AM	ETX, FSELC	Minimum cost path	Multiply transceiver power consumption with transmission time
CAOF	Nodes capabilities, battery level, location	Based on the route metric	Maximum number of seconds a battery can hold
Neighbour-hood metric	ETX, neighbourhood	Set of routes	Load balancing
LB-RPL	Load balance mechanism, link layer communication	Delay DIO	Load balancing

Table 8.1: Comparison of studied energy routing

nodes by distributing the load thus energy consumption in the network.

4. **Path selection** - The studies have different objectives in their path selection, which are to maximise the minimum nodes lifetime [55], to have a network whose nodes deplete at similar speed [28, 76, 55, 18, 113] and to minimise the maximum energy consumption [59, 10]. Despite the differences, the main goal is to consider the nodes' remaining energy or workload thus energy consumption, in deciding the routes to increase the overall network lifetime.

8.7 Conclusion

This chapter proposes the lifetime energy tree algorithm that tries to maximise the minimum lifetime of a node until the tree is found to be improved. The results showed an increase in the network lifetime by 8.3 times more for the improved tree compared to the initial tree of 0.3% lifetime value in the 500 nodes system. It also presents the existing solutions that were proposed to measure the energy consumption used by the nodes. It is important to know the residual energy of the nodes in order to make the decisions, such as to reconstruct the network topology and redirect the nodes to use different routes to prolong the network lifetime. There are two main ways, which are by direct polling and energy estimation, by calculating the radio duty cycle or by the load distribution means. The literature suggests different options that could be taken as the energy consumption measurement.

Chapter 9

Conclusions and Future Research Directions

9.1 Summary

WSNs are widely used in many crucial applications such as in remote environmental monitoring and target tracking, as sensor nodes can easily be deployed in difficult locations. However, WSNs suffer from the sensor nodes' limited hardware and energy capabilities, and the unreliable network environment, which impacts the sensor nodes' performance, and therefore the efficiency of the network.

The work presented in this thesis investigated the benefits of multichannel protocol in ensuring a reliable and energy efficient network. MCRP is presented, which is a decentralised cross-layer protocol with a centralised controller to mitigate the effect of interference without the need to have any knowledge of the channels' occupancy at any location beforehand. This generality enables the sensor nodes to be deployed and select the channel accordingly, depending on the current channel condition at the specific position.

In order to ensure the channel selected has less interference than the current channel or none at all, MCRP provides feedback when a channel is subject to interference using the probing phase. The protocol mitigates the effect of interference by avoiding the affected channels through channel switching processes. It allows better spectrum usage by moving nearby nodes to listen on different channels using a two-hops colouring algorithm.

The performance evaluation showed that MCRP avoids channels with interference, hence greatly reducing loss rate with negligible overheads. By reducing the number of packet losses (hence retransmissions) and increasing the efficiency of spectrum usage, the multichannel system will be more energy-efficient than a single channel protocol over the lifetime of the system's deployment. The centralised controller enables information storing and complicated processing to be undertaken without being restricted by the memory and processing limitations of the sensor nodes. In addition, the condition of the channels can be learned over time, as the information is stored at the controller. This information could be useful for future channel selection. As a result, the time taken in MCRP processes could be reduced if the channel conditions are similar over a period of time.

To further improve the network in addition to the multichannel protocol, MCRP lifetime energy spanning tree algorithm was introduced. The algorithm maximises the minimum node lifetime value by swapping the next hop node to another node from a list of potential nodes that could increase the lifetime of the minimum node. The improved tree is found when there is no other available path with a higher lifetime value. The results of the experiment showed an increase in the network lifetime compared to the initial tree, proving that the network is now more energy balanced than it was previously.

9.2 Future Research Directions

Potential future research directions of the work presented in this thesis can be summarised as follows.

Decentralised protocol: MCRP is currently a decentralised protocol with a centralised controller. The centralised controller is introduced to overcome the limitations of the sensor nodes' hardware in terms of the ability to store and run complicated processes. However, MCRP is fully functional if the controller fails. The centralised controller has the intelligence in interference-free channel selection, where the final decisions are made by the sensor nodes themselves. The controller also stores information about the condition of all nodes' channels. In future work, a fully decentralised protocol could be developed. This eliminates a central controller and the channel selection and decisions are implemented in the nodes themselves.

Hardware independent: TelosB motes are used in this thesis. The sensor nodes have several limitations, as mentioned previously. In order to fully enable a decentralised MCRP protocol to be developed, the sensor nodes have to be able to store all the intelligence. Many of the existing protocols proposed get around this problem by simplifying the multichannel protocol problems by introducing a fixed number of channels, which enables all channels to be used. The logic is to try the channels and retransmit on the next iteration, which will be on a different channel if the transmission failed previously. By making MCRP hardware independent, the protocol can be tested with many recently developed pieces of hardware that are compatible with other hardware that could have better radio coverage and storage to enable a decentralised MCRP.

Mobile networks: MCRP is assumed to be implemented on a static network, allowing new nodes to join, but the nodes' positions are fixed. In future work, a multichannel protocol for mobile networks could be investigated. The nodes should be able to self-configure as the network topology is likely to have frequent changes due to the movement while maintaining high network optimisation. A popular routing technique used in many papers is using a GPS module [126, 129] on the sensor nodes to form a topology or cluster for the communication, to avoid duplication and the nodes falling off the network. However, this comes at the cost of energy and bandwidth, to invoke the geographical location of each sensor node periodically. A multichannel protocol would bring several benefits to mobile networks, as different locations would have different occupancy of channels. MCRP would allow the sensor nodes to switch to different channels accordingly, which could improve the communication. However, a mechanism needs to be introduced to ensure that the nodes could keep track of the nearby nodes' channels. The topology changes would affect the knowledge that the nodes have on the channels to communicate among the nodes.

On-line and off-line property: MCRP is currently an on-line protocol, where the channel decisions are made in real-time to ensure that the communication takes place on a better channel at any location and time. However, in environments such as offices and universities, where the spectrum usage is centrally managed, it is possible to include the off-line property in MCRP, as the interference and usage of channels are typically similar for a period of time, that could be days or weeks. The off-line property refers to enabling MCRP to cache MCRP results on previously successful checked channels for each node. This eliminates the extra packets and time to run MCRP processes. This would allow the nodes to switch to several better channels throughout the day as MCRP has learned the interference pattern for the specific location. MCRP should only allow the offline protocol to be used when the location has no variation in interference from new devices interfering on the previously clear channels. As future work, an algorithm needs to be implemented on MCRP where it decides the amount of time it requires to learn the interference pattern of a location and whether the cached data can be used to allow faster channel switching for the network, which could further improve the overall throughput of the network.

Lifetime energy topology: A more generic approach algorithm is required in order to be able to find near optimal topology. The definition of lifetime used in this thesis only considers the worst-case scenario, where it tries to maximise the minimum node energy as the network becomes non-functional if a node runs out of energy. As future work, the definition of lifetime could be changed. Sensor nodes are cheap and usually densely deployed, therefore it can be assumed that there are several alternative paths to the LPBR if the main node dies. This allows the overall network lifetime to be improved by sacrificing some nodes below a threshold required to maintain the network connectivity. The lifetime energy spanning tree implemented in this thesis only considers changing one node at a time. Changing several nodes might enable a better performing tree to be found. Another option to consider is to allow an individual node to dynamically change its path based on the node's energy consumption in real-time without being instructed by the centralised controller node (LPBR). This enables the path swapping to be done between the node and the surrounding neighbouring nodes without affecting the whole network. However, it might require a long period of time before the final improved tree is found, as this involves each node deciding on the path swapping individually. Thorough research needs to be undertaken, as it could break the whole network if many nodes switch at the same time.

Bibliography

[1] Support of multiple sinks via a virtual root for the RPL routing protocol. *EURASIP Journal on Wireless Communications and Networking*, 2014(1):91, 2014.

[2] Kemal Akkaya and Mohamed Younis. A survey on routing protocols for wireless sensor networks. *Ad hoc networks*, 3(3):325–349, 2005.

[3] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci. Wireless sensor networks: A survey. *Comput. Netw.*, 38(4):393–422, March 2002.

[4] Ian F. Akyildiz, Tommaso Melodia, and Kaushik R. Chowdhury. A survey on wireless multimedia sensor networks. *Comput. Netw.*, 51(4):921–960, March 2007.

[5] B. Al Nahas, S. Duquennoy, V. Iyer, and T. Voigt. Low-power listening goes multichannel. In *2014 IEEE International Conference on Distributed Computing in Sensor Systems (DCOSS)*, pages 2–9, May 2014.

[6] Asaduzzaman and Hyung Yun Kong. Energy efficient cooperative LEACH protocol for wireless sensor networks. *Communications and Networks, Journal of*, 12(4):358–365, Aug 2010.

[7] A.A. Aziz, Y.A. Sekercioglu, P. Fitzpatrick, and M. Ivanovich. A survey on distributed topology control techniques for extending the lifetime of battery powered wireless sensor networks. *Communications Surveys Tutorials, IEEE*, 15(1):121–144, First 2013.

[8] A. Bachir, M. Dohler, T. Watteyne, and K.K. Leung. MAC essentials for wireless sensor networks. *Communications Surveys Tutorials, IEEE*, 12(2):222–248, Second 2010.

[9] Sibbald Barbara. Use computerized systems to cut adverse drug events: report. *CMAJ*, 164(13):1878, June 2001.

[10] A. Barbato, M. Barrano, A. Capone, and N. Figiani. Resource oriented and energy efficient routing protocol for IPv6 wireless sensor networks. In *Online Conference on Green Communications (GreenCom), 2013 IEEE*, pages 163–168, Oct 2013.

[11] Jalel Ben-Othman and Bashir Yahya. Energy efficient and QoS based routing protocol for wireless sensor networks. *J. Parallel Distrib. Comput.*, 70(8):849–857, August 2010.

[12] H. Besbes, G. Smart, D. Buranapanichkit, C. Kloukinas, and Y. Andreopoulos. Analytic conditions for energy neutrality in uniformly-formed wireless sensor networks. *Wireless Communications, IEEE Transactions on*, 12(10):4916–4931, October 2013.

[13] C. A. Boano, Z. He, Y. Li, T. Voigt, M. Zúñniga, and A. Willig. Controllable radio interference for experimental and testing purposes in wireless sensor networks. In *2009 IEEE 34th Conference on Local Computer Networks*, pages 865–872, Oct 2009.

[14] Carlo Alberto Boano, Thiemo Voigt, Nicolas Tsiftes, Luca Mottola, Kay Römer, and Marco Antonio Zúñiga. Making sensornet MAC protocols robust against interference. In *Proceedings of the 7th European Conference on Wireless Sensor Networks*, EWSN'10, pages 272–288, 2010.

[15] Joris Borms, Kris Steenhaut, and Bart Lemmens. Low-overhead dynamic multichannel MAC for wireless sensor networks. In *Proceedings of the 7th European Conference on Wireless Sensor Networks*, EWSN'10, pages 81–96, 2010.

[16] David Braginsky and Deborah Estrin. Rumor routing algorthim for sensor networks. In *Proceedings of the 1st ACM international workshop on Wireless sensor networks and applications*, pages 22–31. ACM, 2002.

[17] Michael Buettner, Gary V. Yee, Eric Anderson, and Richard Han. X-MAC: A short preamble MAC protocol for duty-cycled wireless sensor networks. In *Proceedings of the 4th International Conference on Embedded Networked Sensor Systems*, SenSys '06, pages 307–320, New York, NY, USA, 2006. ACM.

[18] S. Capone, R. Brama, N. Accettura, D. Striccoli, and G. Boggia. An energy efficient and reliable composite metric for RPL organized networks. In *Embedded and Ubiquitous Computing (EUC), 2014 12th IEEE International Conference on*, pages 178–184, Aug 2014.

[19] Alberto Cerpa, Jeremy Elson, Deborah Estrin, Lewis Girod, Michael Hamilton, and Jerry Zhao. Habitat monitoring: Application driver for wireless communications technology. *SIGCOMM Comput. Commun. Rev.*, 31(2 supplement):20–41, April 2001.

[20] Jae-Hwan Chang and L. Tassiulas. Maximum lifetime routing in wireless sensor networks. *Networking, IEEE/ACM Transactions on*, 12(4):609–619, Aug 2004.

[21] L. H. Chang, T. H. Lee, S. J. Chen, and C. Y. Liao. Energy-efficient oriented routing algorithm in wireless sensor networks. In *2013 IEEE International Conference on Systems, Man, and Cybernetics*, pages 3813–3818, Oct 2013.

[22] Yunxia Chen and Qing Zhao. On the lifetime of wireless sensor networks. *Communications Letters, IEEE*, 9(11):976–978, Nov 2005.

[23] Thang Vu Chien, Hung Nguyen Chan, and Thanh Nguyen Huu. A comparative study on operating system for Wireless Sensor Networks. In *2011 International Conference on Advanced Computer Science and Information System (ICACSIS)*, pages 73–78, December 2011.

[24] Maurice Chu, Horst Haussecker, and Feng Zhao. Scalable information-driven sensor querying and routing for ad hoc heterogeneous sensor networks. *International Journal of High Performance Computing Applications*, 16(3):293–313, 2002.

[25] Contiki. Contiki 2.6: Documentation, Jul 2012.

[26] Contiki. Contiki 2.6 The uIP TCP/IP stack, Jul 2012.

[27] S. Dawans, S. Duquennoy, and O. Bonaventure. On link estimation in dense RPL deployments. In *Local Computer Networks Workshops (LCN Workshops), 2012 IEEE 37th Conference on*, pages 952–955, Oct 2012.

[28] D. T. Delaney, L. Xu, and G. M. P. O'Hare. Spreading the load in a tree type routing structure. In *2013 22nd International Conference on Computer Communication and Networks (ICCCN)*, pages 1–7, July 2013.

[29] A. Dunkels, B. Gronvall, and T. Voigt. Contiki - a lightweight and flexible operating system for tiny networked sensors. In *Local Computer Networks, 2004. 29th Annual IEEE International Conference on*, pages 455–462, Nov 2004.

[30] Adam Dunkels. Full tcp/ip for 8-bit architectures. In *Proceedings of the 1st International Conference on Mobile Systems, Applications and Services*, MobiSys '03, pages 85–98, New York, NY, USA, 2003. ACM.

[31] Adam Dunkels. Rime - a lightweight layered communication stack for sensor networks. In *Proceedings of the European Conference on Wireless Sensor Networks (EWSN), Poster/Demo session, Delft, The Netherlands*, January 2007.

[32] Adam Dunkels. Contiki Crash Course, October 2008.

[33] Adam Dunkels. The ContikiMAC radio duty cycling protocol, 2011.

[34] Adam Dunkels, Fredrik Österlind, and Zhitao He. An adaptive communication architecture for wireless sensor networks. In *Proceedings of the 5th International Conference on Embedded Networked Sensor Systems*, SenSys '07, pages 335–349, New York, NY, USA, 2007. ACM.

[35] Adam Dunkels, Fredrik Osterlind, Nicolas Tsiftes, and Zhitao He. Software-based on-line energy estimation for sensor nodes. In *Proceedings of the 4th workshop on Embedded networked sensors*, pages 28–32. ACM, 2007.

[36] Adam Dunkels, Joakim Eriksson, Niclas Finne, and Nicolas Tsiftes. Powertrace: Network-level power profiling for low-power wireless networks. 2011.

[37] Simon Duquennoy, Olaf Landsiedel, and Thiemo Voigt. Let the tree bloom: Scalable opportunistic routing with ORPL. In *Proceedings of the 11th ACM Conference on Embedded Networked Sensor Systems*, SenSys '13, pages 2:1–2:14, 2013.

[38] Simon Duquennoy, Beshr Al Nahas, Olaf Landsiedel, and Thomas Watteyne. Orchestra: Robust mesh networks through autonomously scheduled TSCH. In *Proceed-*

ings of the 13th ACM Conference on Embedded Networked Sensor Systems, SenSys '15, pages 337–350, New York, NY, USA, 2015. ACM.

[39] Amre El-Hoiydi and Jean-Dominique Decotignie. WiseMAC: An ultra low power MAC protocol for multi-hop wireless sensor networks. In *Algorithmic Aspects of Wireless Sensor Networks*, pages 18–31. Springer, 2004.

[40] I.A. Essa. Ubiquitous sensing for smart and aware environments. *Personal Communications, IEEE*, 7(5):47–49, Oct 2000.

[41] R. Fonseca, O. Gnawali, K. Jamieson, P. Levis S. Kim, and A. Woo. TEP 123: The Collection Tree Protocol, Aug 2006.

[42] James M Gilbert and Farooq Balouchi. Comparison of energy harvesting systems for wireless sensor networks. *international journal of automation and computing*, 5(4): 334–347, 2008.

[43] Omprakash Gnawali. The minimum rank with hysteresis objective function, RFC 6719, 2012.

[44] Omprakash Gnawali, Rodrigo Fonseca, Kyle Jamieson, David Moss, and Philip Levis. Collection tree protocol. In *Proceedings of the 7th ACM Conference on Embedded Networked Sensor Systems*, SenSys '09, pages 1–14, 2009.

[45] Wendi Rabiner Heinzelman, Joanna Kulik, and Hari Balakrishnan. Adaptive protocols for information dissemination in wireless sensor networks. In *Proceedings of the 5th annual ACM/IEEE international conference on Mobile computing and networking*, pages 174–185. ACM, 1999.

[46] W.R. Heinzelman, A. Chandrakasan, and H. Balakrishnan. Energy-efficient communication protocol for wireless microsensor networks. In *System Sciences, 2000. Proceedings of the 33rd Annual Hawaii International Conference on*, pages 10 pp. vol.2–, Jan 2000.

[47] J Hui and P Thubert. Compression format for IPv6 datagrams over IEEE 802.15.4-based networks, RFC 6282, 2011.

[48] IEEE. IEEE standard for local and metropolitan area networks–part 15.4: Low-rate wireless personal area networks (LR-WPANs) amendment 1: MAC sublayer. *IEEE*

Std 802.15.4e-2012 (Amendment to IEEE Std 802.15.4-2011), pages 1–225, April 2012.

[49] IEEE. IEEE standard for information technology–telecommunications and information exchange between systems local and metropolitan area networks–specific requirements part 11. *IEEE Std 802.11-2012 (Revision of IEEE Std 802.11-2007)*, pages 1–2793, March 2012.

[50] S. K. A. Imon, A. Khan, M. Di Francesco, and S. K. Das. Energy-efficient randomized switching for maximizing lifetime in tree-based wireless sensor networks. *IEEE/ACM Transactions on Networking*, 23(5):1401–1415, Oct 2015.

[51] Ozlem Durmaz Incel, Lodewijk van Hoesel, Pierre Jansen, and Paul Havinga. MC-LMAC: A multi-channel MAC protocol for wireless sensor networks. *Ad Hoc Netw.*, 9(1):73–94, January 2011. ISSN 1570-8705.

[52] Olle Inganäs and Villy Sundström. Solar energy for electricity and fuels. *Ambio*, 45 (1):15–23, 2016.

[53] Chalermek Intanagonwiwat, Ramesh Govindan, and Deborah Estrin. Directed diffusion: a scalable and robust communication paradigm for sensor networks. In *Proceedings of the 6th annual international conference on Mobile computing and networking*, pages 56–67. ACM, 2000.

[54] FIT IoT-LAB. Building Contiki's tunslip6.

[55] O. Iova, F. Theoleyre, and T. Noel. Improving the network lifetime with energy-balancing routing: Application to RPL. In *Wireless and Mobile Networking Conference (WMNC), 2014 7th IFIP*, pages 1–8, May 2014.

[56] V. Iyer, M. Woehrle, and K. Langendoen. Chrysso - a multi-channel approach to mitigate external interference. In *2011 8th Annual IEEE Communications Society Conference on Sensor, Mesh and Ad Hoc Communications and Networks (SECON)*, pages 449–457, June 2011.

[57] T.R. Jensen and B. Toft. *Graph Coloring Problems*. Wiley Series in Discrete Mathematics and Optimization. Wiley, 2011.

[58] Hongbo Jiang, Shudong Jin, and Chonggang Wang. Prediction or not? an energy-efficient framework for clustering-based data collection in wireless sensor networks. *Parallel and Distributed Systems, IEEE Transactions on*, 22(6):1064–1071, June 2011.

[59] Patrick Olivier Kamgueu, Emmanuel Nataf, Thomas Djotio, and Olivier Festor. Energy-based metric for the routing protocol in low-power and lossy network. In *SENSORNETS*, pages 145–148, 2013.

[60] Youngmin Kim, Hyojeong Shin, and Hojung Cha. Y-MAC: An energy-efficient multi-channel MAC protocol for dense wireless sensor networks. In *Information Processing in Sensor Networks, 2008. IPSN '08. International Conference on*, pages 53–63, April 2008.

[61] Alex King, James Brown, John Vidler, and Utz Roedig. Estimating node lifetime in interference environments. In *Local Computer Networks Conference Workshops (LCN Workshops), 2015 IEEE 40th*, pages 796–803, Oct 2015.

[62] JeongGil Ko, Joakim Eriksson, Nicolas Tsiftes, Stephen Dawson-Haggerty, Jean-Philippe Vasseur, Mathilde Durvy, Andreas Terzis, Adam Dunkels, and David Culler. Beyond Interoperability: Pushing the Performance of Sensor Network IP Stacks. In *Proceedings of the 9th ACM Conference on Embedded Networked Sensor Systems*, SenSys '11, pages 1–11, New York, NY, USA, 2011. ACM.

[63] Lakshman Krishnamurthy, Robert Adler, Phil Buonadonna, Jasmeet Chhabra, Mick Flanigan, Nandakishore Kushalnagar, Lama Nachman, and Mark Yarvis. Design and deployment of industrial sensor networks: Experiences from a semiconductor plant and the north sea. In *Proceedings of the 3rd International Conference on Embedded Networked Sensor Systems*, SenSys '05, pages 64–75, New York, NY, USA, 2005. ACM.

[64] Nest Labs. Nest learning thermostat fact sheet.

[65] Philip Levis, T Clausen, Jonathan Hui, Omprakash Gnawali, and J Ko. RFC 6206: The trickle algorithm, 2011.

[66] Li Li and J.Y. Halpern. Minimum-energy mobile wireless networks revisited. In *Communications, 2001. ICC 2001. IEEE International Conference on*, volume 1, pages 278–283 vol.1, Jun 2001.

[67] X. Li, D. Wang, J. McNair, and J. Chen. Residual energy aware channel assignment in cognitive radio sensor networks. In *2011 IEEE Wireless Communications and Networking Conference*, pages 398–403, March 2011.

[68] Chieh-Jan Mike Liang, Nissanka Bodhi Priyantha, Jie Liu, and Andreas Terzis. Surviving wi-fi interference in low power zigbee networks. In *Proceedings of the 8th ACM Conference on Embedded Networked Sensor Systems*, SenSys '10, pages 309–322, New York, NY, USA, 2010. ACM.

[69] Jun Bum Lim, Beakcheol Jang, and Mihail L. Sichitiu. MCAS-MAC: A multichannel asynchronous scheduled MAC protocol for wireless sensor networks. *Computer Communications*, 56:98 – 107, 2015.

[70] Roman Lim, Federico Ferrari, Marco Zimmerling, Christoph Walser, Philipp Sommer, and Jan Beutel. Flocklab: A testbed for distributed, synchronized tracing and profiling of wireless embedded systems. In *Proceedings of the 12th International Conference on Information Processing in Sensor Networks*, IPSN '13, pages 153–166, New York, NY, USA, 2013. ACM.

[71] Y. Y. Lin and K. F. Ssu. Efficient fault-tolerant backbone construction in tmote sky sensor networks. In *Dependable Computing, 2009. PRDC '09. 15th IEEE Pacific Rim International Symposium on*, pages 319–324, Nov 2009.

[72] S. Lindsey and C.S. Raghavendra. PEGASIS: Power-efficient gathering in sensor information systems. In *Aerospace Conference Proceedings, 2002. IEEE*, volume 3, pages 3–1125–3–1130 vol.3, 2002.

[73] S. Lindsey, C. Raghavendra, and Krishna Sivalingam. Data gathering in sensor networks using the energy*delay metric. In *Parallel and Distributed Processing Symposium., Proceedings 15th International*, pages 2001–2008, April 2001.

[74] An-Feng Liu, Peng-Hui Zhang, and Zhi-Gang Chen. Theoretical analysis of the lifetime and energy hole in cluster based wireless sensor networks. *Journal of Parallel and Distributed Computing*, 71(10):1327 – 1355, 2011.

[75] An-Feng Liu, Peng-Hui Zhang, and Zhi-Gang Chen. Theoretical analysis of the lifetime and energy hole in cluster based wireless sensor networks. *Journal of Parallel and Distributed Computing*, 71(10):1327 – 1355, 2011.

[76] Xinxin Liu, Jianlin Guo, G. Bhatti, P. Orlik, and K. Parsons. Load balanced routing for low power and lossy networks. In *2013 IEEE Wireless Communications and Networking Conference (WCNC)*, pages 2238–2243, April 2013.

[77] Smart Parking Ltd. Parking ideas. *ITS International*, 2015.

[78] A. Manjeshwar and D.P. Agrawal. Apteen: a hybrid protocol for efficient routing and comprehensive information retrieval in wireless. In *Parallel and Distributed Processing Symposium., Proceedings International, IPDPS 2002, Abstracts and CD-ROM*, pages 8 pp–, April 2002.

[79] Arati Manjeshwar and D.P. Agrawal. TEEN: a routing protocol for enhanced efficiency in wireless sensor networks. In *Parallel and Distributed Processing Symposium., Proceedings 15th International*, pages 2009–2015, April 2001.

[80] David Moss and Philip Levis. BoX-MACs: Exploiting physical and link layer boundaries in low-power networking. *Computer Systems Laboratory Stanford University*, pages 116–119, 2008.

[81] Emmanuel Nataf and Olivier Festor. Accurate online estimation of battery lifetime for wireless sensors network. In *Proceedings of the 2nd International Conference on Sensor Networks*, pages 59–64, 2013.

[82] N. Nordin, R. G. Clegg, and M. Rio. Multichannel cross-layer routing for sensor networks. In *2016 23rd International Conference on Telecommunications (ICT)*, pages 1–6, May 2016.

[83] N. Noury, T. Herve, V. Rialle, G. Virone, E. Mercier, G. Morey, A. Moro, and T. Porcheron. Monitoring behavior in home using a smart fall sensor and position sensors. In *Microtechnologies in Medicine and Biology, 1st Annual International, Conference On. 2000*, pages 607–610, 2000.

[84] U.S. Department of Commerce. Flood warning systems manual, April 2012.

[85] F. Osterlind, A. Dunkels, J. Eriksson, N. Finne, and T. Voigt. Cross-level sensor network simulation with COOJA. In *Local Computer Networks, Proceedings 2006 31st IEEE Conference on*, pages 641–648, Nov 2006.

[86] N.A. Pantazis, S.A. Nikolidakis, and D.D. Vergados. Energy-efficient routing protocols in wireless sensor networks: A survey. *Communications Surveys Tutorials, IEEE*, 15(2):551–591, Second 2013.

[87] E.M. Petriu, Nicolas D. Georganas, D.C. Petriu, D. Makrakis, and V.Z. Groza. Sensor-based information appliances. *Instrumentation Measurement Magazine, IEEE*, 3(4):31–35, Dec 2000.

[88] M. Petrova, Lili Wu, P. Mahonen, and J. Riihijarvi. Interference measurements on performance degradation between colocated IEEE 802.11g/n and IEEE 802.15.4 networks. In *Networking, 2007. ICN '07. Sixth International Conference on*, pages 93–93, April 2007.

[89] Joseph Polastre, Jason Hill, and David Culler. Versatile low power media access for wireless sensor networks. In *Proceedings of the 2nd international conference on Embedded networked sensor systems*, pages 95–107. ACM, 2004.

[90] A. Redondi, D. Buranapanichkit, M. Cesana, M. Tagliasacchi, and Y. Andreopoulos. Energy consumption of visual sensor networks: Impact of spatio-temporal coverage. *Circuits and Systems for Video Technology, IEEE Transactions on*, 24(12):2117–2131, Dec 2014.

[91] J Romkey. A nonstandard for transmission of IP datagrams over serial lines: SLIP, RFC 1055, 1988.

[92] Narayanan Sadagopan, Bhaskar Krishnamachari, and Ahmed Helmy. The ACQUIRE mechanism for efficient querying in sensor networks. In *Sensor Network Protocols and Applications, 2003. Proceedings of the First IEEE. 2003 IEEE International Workshop on*, pages 149–155. IEEE, 2003.

[93] Ltd. Samsung Electronics Co. Smartthings future living report.

[94] Paolo Santi. Topology control in wireless ad hoc and sensor networks. *ACM Comput. Surv.*, 37(2):164–194, June 2005.

[95] Curt Schurgers and Mani B Srivastava. Energy efficient routing in wireless sensor networks. In *Military Communications Conference, 2001. MILCOM 2001. Communications for Network-Centric Operations: Creating the Information Force. IEEE*, volume 1, pages 357–361. IEEE, 2001.

[96] W.K.G. Seah, Zhi Ang Eu, and H. Tan. Wireless sensor networks powered by ambient energy harvesting (WSN-HEAP) - survey and challenges. In *Wireless Communication, Vehicular Technology, Information Theory and Aerospace Electronic Systems Technology, 2009. Wireless VITAE 2009. 1st International Conference on*, pages 1–5, May 2009.

[97] M. Sha, G. Hackmann, and C. Lu. Real-world empirical studies on multi-channel reliability and spectrum usage for home-area sensor networks. *IEEE Transactions on Network and Service Management*, 10(1):56–69, March 2013.

[98] B. Sharkawy, A. Khattab, and K. M. F. Elsayed. Fault-tolerant RPL through context awareness. In *Internet of Things (WF-IoT), 2014 IEEE World Forum on*, pages 437–441, March 2014.

[99] Jang-Ping Sheu, Kun-Ying Hsieh, and Po-Wen Cheng. Design and implementation of mobile robot for nodes replacement in wireless sensor networks. *J. Inf. Sci. Eng.*, 24(2):393–410, 2008.

[100] Gyula Simon, Miklós Maróti, Ákos Lédeczi, György Balogh, Branislav Kusy, András Nádas, Gábor Pap, János Sallai, and Ken Frampton. Sensor network-based countersniper system. In *Proceedings of the 2Nd International Conference on Embedded Networked Sensor Systems*, SenSys '04, pages 1–12, New York, NY, USA, 2004. ACM. ISBN 1-58113-879-2.

[101] Shio Kumar Singh, MP Singh, DK Singh, et al. Routing protocols in wireless sensor networks–a survey. *International Journal of Computer Science & Engineering Survey (IJCSES) Vol*, 1:63–83, 2010.

[102] Lanny Sitanayah, Cormac J. Sreenan, and Szymon Fedor. A cooja-based tool for maintaining sensor network coverage requirements in a building. In *Proceedings of the 11th ACM Conference on Embedded Networked Sensor Systems*, SenSys '13, pages 70:1–70:2, 2013.

[103] A. Sivanantha, B. Hamdaoui, M. Guizani, Xiuzhen Cheng, and T. Znati. EM-MAC: An energy-aware multi-channel MAC protocol for multi-hop wireless networks. In *Wireless Communications and Mobile Computing Conference (IWCMC), 2012 8th International*, pages 1159–1164, Aug 2012.

[104] Tmote Sky. Ultra low power ieee 802.15. 4 compliant wireless sensor module. *Moteiv Corporation*, 2006.

[105] K. Sohrabi, J. Gao, V. Ailawadhi, and G.J. Pottie. Protocols for self-organization of a wireless sensor network. *Personal Communications, IEEE*, 7(5):16–27, Oct 2000.

[106] Ridha Soua and Pascale Minet. Multichannel assignment protocols in wireless sensor networks: A comprehensive survey. *Pervasive and Mobile Computing*, 16, Part A:2 – 21, 2015.

[107] L. Stabellini. Energy-aware channel selection for cognitive wireless sensor networks. In *Wireless Communication Systems (ISWCS), 2010 7th International Symposium on*, pages 892–896, Sept 2010.

[108] Streetline. Parking data and analytics: What can your parking spaces tell you?

[109] Lei Tang, Yanjun Sun, O. Gurewitz, and D.B. Johnson. PW-MAC: An energy-efficient predictive-wakeup mac protocol for wireless sensor networks. In *INFOCOM, 2011 Proceedings IEEE*, pages 1305–1313, April 2011.

[110] Crossbow Technology. TelosB - TelosB mote platform. Document Part Number: 6020-0094-01 Rev B.

[111] Luigi Alfredo Grieco Thomas Watteyne, Maria Rita Palattella. Using IEEE802.15.4e time-slotted channel hopping (TSCH) in an internet of things (IoT): Problem statement, May 2015.

[112] Pascal Thubert. Objective function zero for the routing protocol for low-power and lossy networks (RPL), RFC 6552, 2012.

[113] D. Todolí-Ferrandis, S. Santonja-Climent, V. Sempere-Payá, and J. Silvestre-Blanes. RPL routing in a real life scenario with an energy efficient objective function. In *Telecommunications Forum Telfor (TELFOR), 2015 23rd*, pages 285–288, Nov 2015.

[114] Nicolas Tsiftes, Joakim Eriksson, Niclas Finne, Fredrik Osterlind, Joel Höglund, and Adam Dunkels. A framework for low-power IPv6 routing simulation, experimentation, and evaluation. In *Proceedings of the ACM SIGCOMM 2010 Conference*, SIGCOMM '10, pages 479–480, New York, NY, USA, 2010.

[115] Tsvetko Tsvetkov. RPL: IPv6 routing protocol for low power and lossy networks. *Sensor Nodes–Operation, Network and Application (SN)*, 59:2, 2011.

[116] L.F.W. van Hoesel and P.J.M. Havinga. A lightweight medium access protocol (LMAC) for wireless sensor networks: Reducing preamble transmissions and transceiver state switches. In *1st International Workshop on Networked Sensing Systems, INSS 2004*, pages 205–208, Tokyo, Japan, 2004. Society of Instrument and Control Engineers (SICE).

[117] J Vasseur, M Kim, K Pister, N Dejean, and D Barthel. Routing metrics used for path calculation in low power and lossy networks, 2012.

[118] Z.G. Wan, Y.K. Tan, and C. Yuen. Review on energy harvesting and energy management for sustainable wireless sensor networks. In *Communication Technology (ICCT), 2011 IEEE 13th International Conference on*, pages 362–367, Sept 2011.

[119] Thomas Watteyne, Ankur Mehta, and Kris Pister. Reliability through frequency diversity: Why channel hopping makes sense. In *Proceedings of the 6th ACM Symposium on Performance Evaluation of Wireless Ad Hoc, Sensor, and Ubiquitous Networks*, pages 116–123, 2009.

[120] Geoffrey Werner-Allen, Konrad Lorincz, Matt Welsh, Omar Marcillo, Jeff Johnson, Mario Ruiz, and Jonathan Lees. Deploying a wireless sensor network on an active volcano. *IEEE Internet Computing*, 10(2):18–25, March 2006. ISSN 1089-7801.

[121] T Winter, P Thubert, T Clausen, J Hui, R Kelsey, P Levis, K Pister, R Struik, and J Vasseur. RPL: IPv6 routing protocol for low power and lossy networks, RFC 6550, 2012.

[122] Jie Wu, Ming Gao, and I. Stojmenovic. On calculating power-aware connected dominating sets for efficient routing in ad hoc wireless networks. In *Parallel Processing, 2001. International Conference on*, pages 346–354, Sept 2001.

[123] Yafeng Wu, J.A. Stankovic, Tian He, and Shan Lin. Realistic and efficient multi-channel communications in wireless sensor networks. In *IEEE INFOCOM 2008. The 27th Conference on Computer Communications*, April 2008.

[124] Yafeng Wu, Kin Sum Liu, John A. Stankovic, Tian He, and Shan Lin. Efficient multichannel communications in wireless sensor networks. *ACM Trans. Sen. Netw.*, 12(1):3:1–3:23, March 2016.

[125] Yueshi Wu and Mihaela Cardei. Multi-channel and cognitive radio approaches for wireless sensor networks. *Computer Communications*, 94:30 – 45, 2016.

[126] Ya Xu, John Heidemann, and Deborah Estrin. Geography-informed energy conservation for ad hoc routing. In *Proceedings of the 7th Annual International Conference on Mobile Computing and Networking*, MobiCom '01, pages 70–84, New York, NY, USA, 2001. ACM.

[127] Jennifer Yick, Biswanath Mukherjee, and Dipak Ghosal. Wireless sensor network survey. *Comput. Netw.*, 52(12):2292–2330, August 2008.

[128] O. Younis and Sonia Fahmy. HEED: a hybrid, energy-efficient, distributed clustering approach for ad hoc sensor networks. *Mobile Computing, IEEE Transactions on*, 3 (4):366–379, Oct 2004.

[129] Yan Yu, Ramesh Govindan, and Deborah Estrin. Geographical and energy aware routing: A recursive data dissemination protocol for wireless sensor networks. Technical report, Technical report ucla/csd-tr-01-0023, UCLA Computer Science Department, 2001.

[130] Yi-hua Zhu, Wan-deng Wu, Jian Pan, and Yi-ping Tang. An energy-efficient data gathering algorithm to prolong lifetime of wireless sensor networks. *Comput. Commun.*, 33(5):639–647, March 2010.

[131] Yi-hua Zhu, Wan-deng Wu, and Victor C. M. Leung. Energy-efficient tree-based message ferrying routing schemes for wireless sensor networks. *Mobile Networks and Applications*, 16(1):58–70, 2011.